

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh a implementace databázové aplikace pro projekční kancelář
Design and Implementation of Database Application for the Project Office

Student:

Pavel Bittner

Vedoucí bakalářské práce:

Ing. Vítězslav Novák, Ph.D.

Ostrava 2015

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra systémového inženýrství

Zadání bakalářské práce

Student: **Pavel Bittner**

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: 6209R025 Systémové inženýrství a informatika

Téma: **Návrh a implementace databázové aplikace pro projekční kancelář**
Design and Implementation of Database Application for the Project Office

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska tvorby databázových aplikací
3. Analýza současného stavu ve firmě
4. Návrh a implementace databázové aplikace
5. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků bakalářské práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. 392 s. ISBN 978-80-251-3289-0.

PÍSEK, Slavoj. *Access 2010: podrobný průvodce*. Praha: Grada Publishing, 2011. 160 s. ISBN 978-80-247-3653-2.

SHEPHERD, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. 397 s. ISBN 978-80-251-3686-7.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 21.11.2014

Datum odevzdání: 07.05.2015



doc. Ing. Jana Hančlová, CSc.
vedoucí katedry



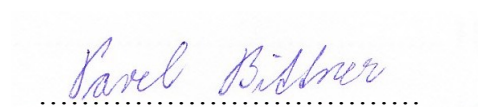
prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Místopřísežné prohlášení o samostatném vypracování bakalářské práce

„Prohlašuji, že jsem celou práci, včetně všech příloh, vypracoval samostatně“.

Zároveň bych na tomto místě chtěl poděkovat zejména svému vedoucímu bakalářské práce Ing. Vítězslavu Novákovi, Ph.D. za jeho cenné rady a připomínky ohledně mé bakalářské práce. Dále pak Ing. Daně Kožušníkové za poskytnutí informací a materiálů potřebných k vypracování mé bakalářské práce.

Datum odevzdání bakalářské práce: 7. 5. 2015



Pavel Bittner

Obsah

1	Úvod	5
2	Teoretická východiska tvorby databázových aplikací	6
2.1	Databázový systém	6
2.1.1	Pojem databáze	6
2.1.2	Systém řízení báze dat	6
2.1.3	Databázová aplikace	7
2.1.4	Uživatelé	7
2.2	Databázové (logické) modely	7
2.2.1	Hierarchický model	7
2.2.2	Síťový model	8
2.2.3	Relační model	8
2.2.4	Objektově orientovaný model	9
2.2.5	Objektově relační model	9
2.3	Datové modelování	9
2.3.1	Konceptuální model	10
2.3.2	Logický model	13
2.3.3	Fyzický model	15
2.4	SQL	16
2.4.1	Data Definition Language (DDL)	16
2.4.2	Data Query Language (DQL)	17
2.4.3	Data Manipulation Language (DML)	17
2.4.4	Data Control Language (DCL)	17
2.4.5	Příkazy řízení transakcí	17
2.5	Microsoft Access 2010	18
2.5.1	Relace (tabulky)	18
2.5.2	Základní datové typy	19
2.5.3	Vlastnosti základních datových typů	20

2.5.4	Vztahy	21
2.5.5	Formuláře	21
2.5.6	Tiskové sestavy	22
2.5.7	Filtry	23
2.5.8	Dotazy	24
2.5.9	Makra	24
2.5.10	Visual Basic for Application (VBA)	25
3	Analýza současného stavu ve firmě	26
3.1	Požadavky na aplikaci	26
4	Návrh a implementace databázové aplikace	27
4.1	Konceptuální model	27
4.1.1	Popis entit	27
4.2	Logický model	30
4.3	Aplikace Microsoft Access	31
4.3.1	Tabulky	31
4.3.2	Vztahy	35
4.3.3	Formuláře	36
4.3.4	Sestavy	44
4.3.5	Dotazy	45
4.3.6	Dokončení aplikace	46
4.4	Nasazení aplikace	46
5	Závěr	47
	Seznam použité literatury	48
	Seznam zkratk	49
	Prohlášení o využití výsledků bakalářské práce	
	Seznam příloh	

1 Úvod

Neustálý a rychlý vývoj technologií a především vývoj v oblasti informačních technologií je důvodem zvolení tématu návrhu databázové aplikace. Souvisí s autorovým zájmem o tuto oblast. V každém podniku se používá určitá forma databáze a nezáleží na velikosti podniku nebo druhu jeho činnosti. Autor proto doufá, že mu nabyté znalosti v této oblasti budou v blízké budoucnosti užitečné.

Cílem bakalářské práce je navrhnout a implementovat databázovou aplikaci v prostředí Microsoft Access 2010 pro projekční kancelář. Účelem aplikace je zefektivnit a zrychlit proces zaznamenávání a vyhledávání dat důležitých pro projekční kancelář. Aplikace bude sloužit k průběžnému vedení informací o zákaznících, projektech, službách a zpracovatelích projekční kanceláře. Finální verze databázové aplikace by měla být přehledná a intuitivní, aby se dala pohodlně obsluhovat.

Bakalářská práce je členěna do pěti hlavních částí.

V první části *Úvodu* je popsán cíl bakalářské práce a členění hlavních kapitol se stručným popisem.

V části *Teoretická východiska tvorby databázových aplikací* se práce zabývá teorií spojenou s tvorbou databázových aplikací, která je teoretickým základem pro tvorbu této aplikace. Dále jsou zde vysvětleny základní pojmy související s tvorbou databázových aplikací v prostředí Microsoft Access 2010.

Třetí část s názvem *Analýza současného stavu ve firmě* se zabývá popisem a analýzou současného stavu ve firmě. Dále je zde popsáno, co by měla výsledná aplikace splňovat.

Stěžejní částí je kapitola *Návrh a implementace databázové aplikace*. Kapitola je zaměřena na návrh a vytvoření samotné databázové aplikace v programu Microsoft Access. Nejdříve je proveden návrh konceptuálního modelu, poté návrh logického modelu. Nakonec jsou výstupy logického návrhu převedeny do fyzického návrhu aplikace v prostředí Microsoft Access. Dále je v této kapitole podrobně popsána výsledná aplikace. Jsou zde popsány formuláře, které slouží k ovládání aplikace, sestavy a dotazy. Nakonec je popsána implementace výsledné aplikace.

Poslední částí je kapitola *Závěr*, ve které je posouzena a zhodnocena celková práce a její jednotlivé cíle.

2 Teoretická východiska tvorby databázových aplikací

2.1 Databázový systém

Databázový systém se skládá ze čtyř komponent:

- databáze,
- systému řízení báze dat,
- databázové aplikace,
- uživatelů.

2.1.1 Pojem databáze

Co si lze představit pod pojmem databáze? Tento pojem lze definovat vícero způsoby a z tohoto důvodu také dochází k tomu, že každý z autorů tento pojem definuje trochu jinak.

V nejobecnějším případě lze databázi definovat jako kolekci souvisejících záznamů, které obsahují vlastní popis. Vlastní popis znamená, že popis struktury databáze je součástí samotné databáze. (Kroenke a Auer, 2015)

Databáze je sdílená kolekce logicky souvisejících dat i s popisem své datové struktury, organizovaná pro optimální manipulaci s perzistentními daty a získávání informací pro potřeby informačního systému. (Novák, 2014)

2.1.2 Systém řízení báze dat

Systém řízení báze dat (SŘBD) z anglického database management system (DBMS) označuje software, který zajišťuje práci s databází, tvoří rozhraní mezi aplikačními programy a uloženými daty. Úkolem systému řízení báze dat je vytváření, zpracovávání a správa databáze. (Kroenke a Auer, 2015)

Existuje mnoho systémů řízení báze dat, jsou to zejména komerční produkty, které se pořízují formou licence. Mezi ty nejrozšířenější patří:

- DB2 společnosti IBM,
- Oracle Database společnosti Oracle Corporation,
- Microsoft SQL Server,
- MySQL,
- Microsoft Access.

2.1.3 Databázová aplikace

Databázová aplikace je software, který slouží jako prostředník mezi systémem řízení báze dat a uživatelem. Aplikaci tvoří formuláře a sestavy, se kterými pracuje uživatel. Na základě uživatelských požadavků databázová aplikace komunikuje se SŘBD tak, že zasílá SQL (Structured Query Language) příkazy.

2.1.4 Uživatelé

Uživatelé využívají databázové aplikace a pomocí formulářů zadávají a načítají data, vytvářejí dotazy a na jejich základě vytvářejí tiskové sestavy.

2.2 Databázové (logické) modely

Databázový model slouží k modelování struktury databáze. Vyjadřuje logické uspořádání dat v databázi. V průběhu vývoje databází vzniklo několik druhů těchto modelů:

- hierarchický model,
- síťový model,
- relační model,
- objektový model.

2.2.1 Hierarchický model

Jedná se o nejstarší model, vznik tohoto modelu je spojen se systémem IMS společnosti IBM. První verze byla spuštěna v roce 1968. (Kaluža a Kalužová, 2012)

Model je organizován do stromové struktury. Vztahy v tomto modelu jsou založeny na terminologii rodič – potomek. To znamená, že tabulka rodiče může odpovídat jedné nebo více tabulkám potomků. Zároveň platí, že každému potomku může odpovídat pouze jedna tabulka rodiče. Propojení tabulek je zajištěno ukazateli na tyto tabulky (šipkami). Při přístupu k datům je důležité, aby uživatel znal stromovou strukturu databáze, protože přístup k datům probíhá přes kořenovou tabulku. Uživatel postupně prochází její potomky, až se dostane k hledaným datům.

Výhodou tohoto modelu je rychlost vyhledávání dat, což je zapříčiněno explicitním propojením tabulek. Další výhoda plyne z logiky tohoto modelu, které se říká referenční integrita. Pokud dojde k vymazání záznamu rodiče, dojde pak i k vymazání všech propojených záznamů potomků.

Mezi nevýhody tohoto modelu patří situace, kdy je potřeba vložit záznam do tabulky potomka, kterému však neodpovídá žádný záznam v tabulce rodiče. Další nevýhodou tohoto modelu je redundance dat. (Hernandez, 2006)

2.2.2 Sítový model

Sítový model souvisí s hierarchickým modelem, jedná se o jeho zobecnění. Poprvé byl definován v roce 1971. V současnosti se již tento model nevyužívá a jeho vývoj byl ukončen. (Šimonová a Panuš, 2007)

Pojmy užívané v souvislosti se sítovým modelem jsou především uzly a množinové struktury. Uzel vyjadřuje soubor záznamů a vztahy mezi uzly vyjadřuje množinová struktura. Terminologie rodič – potomek z hierarchického modelu byla nahrazena pojmy vlastník - člen. Došlo k vylepšení, kdy uzel vlastník může být v relaci k jednomu nebo více záznamům uzlů člen. Uzel člen musí být ve vztahu pouze k jednomu uzlu vlastník. Zároveň platí, že uzel člen nemůže existovat bez vztahu s uzlem vlastník, kdežto uzel vlastník může existovat bez vztahu s uzlem člen.

Za výhody sítového modelu jsou označovány především rychlost vyhledávání dat v databázi, která spočívá v možnosti procházení databáze pomocí množinových struktur. Další výhodou je možnost vytváření komplexních dotazů, které nebylo možné do takové míry vytvářet v hierarchickém modelu.

Jako hlavní nevýhoda tohoto modelu je označována nutná znalost struktury databáze, aby mohl uživatel efektivně pracovat s databází. Další nevýhoda modelu nastává, když je potřeba změnit strukturu databáze a tím zároveň upravit množinovou strukturu databáze, protože je pak nutno změnit aplikační programy používající tyto množinové struktury. (Hernandez, 2006)

2.2.3 Relační model

Jedná se o dnes nejrozšířenější model. Relační model je výsledek zkoumání, kdy se řešily nedostatky předešlých modelů. První relační databázi představil v roce 1969 Dr. Edgar F. Codd z laboratoře společnosti IBM. (Kaluža a Kalužová, 2012), (Hernandez, 2006)

Data jsou v relačních databázích ukládána v relacích, které se uživatelům jeví jako dvourozměrné tabulky. Každá relace je složena ze záznamů (n-tic), které jsou zaznamenány jako řádek tabulky a polí (atributů), které jsou zaznamenány jako sloupec tabulky. Primární klíč obsahující unikátní hodnotu je důvod, proč nezáleží na fyzickém uspořádání záznamů či polí. Každý takovýto záznam je jednoznačně identifikován pomocí primárního klíče. Mezi relacemi

jsou definovány vztahy. Pomocí vztahů se vytváří spojení mezi dvěma relacemi, které spolu logicky souvisí.

Pomocí jazyka SQL, který je podrobněji popsán v části 2.4, lze databázi vytvářet, modifikovat, spravovat či vytvářet dotazy. Tento jazyk je obvykle implementován do databázového systému. Není vždy nutno znát jazyk SQL, pomocí grafických editorů lze vytvářet SQL příkazy na databázové systémy.

Výhoda modelu spočívá ve snadném upravování struktury databáze. Automaticky je zabudována víceúrovňová integrita, která zajišťuje, že záznamy nebudou duplicitní a detekuje chybějící hodnoty primárních klíčů. Další výhodou je snadné vyhledávání dat z jedné nebo více tabulek, které jsou spolu v relaci. (Hernandez, 2006)

2.2.4 Objektově orientovaný model

Objektově orientovaný model je postaven na objektově orientovaných programovacích jazycích (C++, ASP, Java), což znamená, že nese jejich charakteristiky a výhody. Přenesení vývoje databáze do syntaxe programovacího jazyka je uváděna jako jedna z největších výhod, protože programový kód lze lépe pochopit. S tímto i souvisí snadná a efektivní úprava této databáze. (Šimonová a Panuš, 2007)

2.2.5 Objektově relační model

Jedná se o rozšíření relačního modelu přidáním nových objektů do relačního systému. Jedná se hlavně o třídy, zapouzdření a dědičnost. Cílem tohoto modelu je umožnění práce se složitějšími typy dat jako video či audio záznamy, které relační model do takové míry nepodporuje. Tento model je stále ve vývoji a odborníci diskutují, jak by měla konečná podoba tohoto modelu vypadat. (Hernandez, 2006), (Šimonová a Panuš, 2007)

2.3 Datové modelování

Model systému je vytvářen na základě reality pomocí vhodných metod, postupů a nástrojů. Jako základní princip modelování se uvádí nejdříve vytvoření modelu systému a až poté systém samotný. Je nutno dodat, že model systému realitu do určité míry zjednodušuje a abstrahuje od některých méně podstatných skutečností. Je to z důvodu, že ne všechny tyto skutečnosti jsou podstatné a zároveň není možné, aby vytvořený model reflektoval realitu do všech detailů. Proto je v modelu systému zahrnuto jen to podstatné a důležité. (Šimonová a Panuš, 2007)

Postup datového modelování se dělí do různých fází vývoje, kdy se na data pohlíží různým způsobem. Tyto fáze se dělí na:

- konceptuální model,
- logický model,
- fyzický model.

2.3.1 Konceptuální model

První úroveň datového modelování je konceptuální model. Tento model vytváří popis systému. Nezávisí přitom, nad jakou technologií bude systém nakonec vytvořen, a jak bude implementován. (Šimonová a Panuš, 2007)

Samotný popis systému se většinou provádí pomocí grafického vyjádření. Základní metodou k tomu je vytvoření ER diagramu (Entity-Relationship), který používá různé grafické prvky, jako jsou entity, vztahy, atributy, domény a klíče. (Kaluža a Kalužová, 2012)

2.3.1.1 Entita

Entita představuje reálný objekt, který má být sledován, a mají být o něm uchovávány informace. Takový objekt má být schopen nezávislé existence a být jednoznačně rozlišitelný od ostatních objektů. Příklad entity je například zaměstnanec, automobil nebo objednávka. (Šimonová a Panuš, 2007)

2.3.1.2 Vztahy

Mezi jednotlivými entitami lze definovat vztahy. Jako nejužívanější typ vztahu je udáván asociativní vztah. Vedle vztahu asociace se ještě používá vztah generalizace (generický vztah). Vztah asociace představuje informaci jedné nebo více entit například „*je zaměstnán v*“. Každý vztah asociace je charakterizován třemi charakteristikami: stupněm, kardinalitou a volitelností. (Kaluža a Kalužová, 2012)

Stupeň představuje počet entit ve vztahu. Dělí se na vztahy unární, binární, ternární atd. Unární vztah znamená, že entita má vztah sama se sebou tzv. rekurzivní vztah. Vztahy druhého stupně tzv. binární představují vztah mezi dvěma entitami. Analogicky vztah třetího stupně se nazývá ternární a představuje vztah mezi třemi entitami. (Kroenke a Auer, 2015), (Kaluža a Kalužová, 2012)

Kardinalita představuje maximální a minimální počet výskytů entity v určitém vztahu. Dělí se na tři typy vztahů: 1:1, 1:N, N:M.

- Vztah 1:1 vyjadřuje, kdy záznam v první tabulce je ve vztahu přesně s jedním záznamem z druhé tabulky a záznam druhé tabulky je ve vztahu přesně s jedním záznamem první tabulky.
- Vztah 1:N je nejčastěji užívaným typem vztahu. Tento vztah vyjadřuje, kdy záznam v první tabulce je ve vztahu k několika záznamům druhé tabulky, ale záznam v druhé tabulce může být pouze k jednomu záznamu v první tabulce.
- Dvě tabulky mají vztah M:N, když záznam první tabulky může být ve vztahu k několika záznamům druhé tabulky a jeden záznam druhé tabulky může být ve vztahu s několika záznamy první tabulky. (Hernandez, 2006), (Šimonová a Panuš, 2007)

Volitelnost je třetí charakteristika vztahu asociace. Dělí se na povinnou a volitelnou. Vypovídá o tom, zda musí existovat záznam v druhé tabulce před tím, než je vložen záznam do první tabulky. U povinné musí existovat záznam v druhé tabulce, aby mohl být vložen záznam do první tabulky. U volitelné toto neplatí, a proto nemusí existovat záznam ve druhé tabulce, aby mohl být vložen záznam do první tabulky. (Hernandez, 2006)

2.3.1.3 Atributy

Atributy popisují vlastnosti dané entity. Může se jednat například o jméno či příjmení. Do každého atributu jsou ukládány konkrétní hodnoty, které s ním souvisí. (Kaluža a Kalužová, 2012)

2.3.1.4 Doména

Jako doména se rozumí množina přípustných hodnot přiřazená jednomu nebo více atributům. (Šimonová a Panuš, 2007)

2.3.1.5 Klíče

Klíč je jednoznačný identifikátor entity. Jedná se tedy o nejmenší možnou množinu atributů, které jednoznačně identifikují výskyt dané entity. Klíče se dělí na jednoznačné a složené. O klíči se říká, že je jednoznačný, jestliže se skládá z jednoho atributu. Pokud se skládá z více než jednoho atributu, pak se jedná o klíč složený. (Kaluža a Kalužová, 2012)

O kandidátní klíč se jedná, jestliže klíč jednoznačně identifikuje záznam dané entity. Každá tabulka musí obsahovat minimálně jeden kandidátní klíč. Každý kandidátní klíč musí splňovat určité vlastnosti.

- Kandidátní klíč musí obsahovat jedinečné hodnoty. Z této vlastnosti vyplývá, že není možné, aby byly v tabulce duplicitní záznamy. Tato vlastnost pomáhá předcházet duplicitním záznamům.
- Kandidátní klíč nesmí obsahovat hodnoty null, protože hodnota null představuje chybějící nebo nulovou hodnotu. Identifikace záznamu by pak nebyla možná.
- Kandidátní klíč nesmí zapříčinit únik informací. Z této vlastnosti vyplývá, že není vhodné používat například hesla jako kandidátní klíč.
- Hodnota kandidátního klíče není ani částečně volitelná. Jinak by to znamenalo, že část klíče může obsahovat hodnotu null, a to není možné. Tento problém nastává především, když se kandidátní klíč skládá z více než jednoho pole.
- Kandidátní klíč se skládá z nejmenšího možného počtu polí, která zajišťují jedinečnost.
- Hodnota kandidátního klíče musí být v rámci celé tabulky jedinečná.
- Hodnota kandidátního klíče musí jednoznačně identifikovat hodnoty všech polí daného záznamu. Tato vlastnost zajišťuje, že přístup k polím tabulky je zabezpečen pomocí jediného klíče.
- Hodnota kandidátního klíče se mění jen ve výjimečných situacích. (Hernandez, 2006)

Primární klíč se určí tak, že z množiny kandidátních klíčů se vybere jeden. Tento klíč jednoznačně identifikuje výskyt entity. Obecně platí, že každá tabulka musí mít právě jeden primární klíč a všechny klíče v databázi musí být jedinečné. Z ostatních kandidátních klíčů, které nebyly určeny jako primární klíče, se stávají alternativní klíče. (Kroenke a Auer, 2015), (Kaluža a Kalužová, 2012)

2.3.1.6 Postup vytvoření konceptuálního modelu

1. V rámci prvního kroku vymezení struktury entit, se graficky vyznačí struktura entit a pojmenují se.
2. V dalším kroku se entitám přiřadí atributy. Toto se provádí pomocí vytvořeného seznamu atributů, který se přiřadí k jednotlivým entitám.
3. V následujícím kroku se definují vztahy mezi entitami, určí se kardinalita a volitelnost.
4. Poslední krok souvisí s integrací dílčích částí modelu. Tento krok je důležitý u rozsáhlejších projektů, kdy je potřeba integrovat několik dílčích konceptuálních modelů do jednoho. Uvádí se, že nejrozsáhlejší způsob integrace je tzv. binární přístup, který postupně integruje dvojice dílčích konceptů. (Kaluža a Kalužová, 2012)

2.3.2 Logický model

Druhý krok datového modelování je vytvoření logického modelu. Vychází z výstupů konceptuálního modelu, který měl zachytit všechno to podstatné, co se musí v rámci databáze sledovat, a převádí fungování modelu do některého databázového modelu. Možné druhy databázových modelů jsou popsány v kapitole 2.2. Nejrozšířenějším logickým modelem v dnešní době je relační databázový model, a proto je v této části práce detailněji rozebrán.

Relační model je postaven na principech odvozených z teorie množin a predikátové logiky. Model popisuje, jak definovat strukturu dat, jak tato data ochránit (integrita dat), a operace související s manipulací nad těmito daty. Relační databázový model byl postaven na těchto myšlenkách:

- data jsou chápána jako relace,
- relační model dat důsledně odděluje data od jejich implementace,
- při manipulaci s daty se není nutno zajímat o přístupové mechanismy k datům v relacích,
- manipulace s daty je zajištěna relačním kalkulem a relační algebrou,
- k omezení redundance dat se používá tzv. normalizace relací. (Šimonová a Panuš, 2007)

2.3.2.1 Relace

Relací se podle (Date, 2003) rozumí dvourozměrná datová struktura tvořená záhlavím relace a tělem relace. Kdy záhlaví je množina dvojic (A_i, D_i) , kde atribut A_i je přiřazen právě jedné doméně D_i , pro $i = 1, 2, \dots, n$. Tělo relace je tvořeno množinou n -tic (A_i, v_{ji}) , kde A_i je atribut je i -tý atribut a dále v_{ji} je j -tá hodnota z domény D_i pro $j = 1, 2, \dots, m$, kde m je počet n -tic v množině; m je kardinalitou a n stupněm relace. Mezi základní vlastnosti relací patří:

- neexistence duplicitních n -tic,
- libovolné pořadí n -tic a atributů,
- nerozložitelnost hodnot atributů. (Kaluža a Kalužová, 2012)

2.3.2.2 Normalizace

Normalizace je proces, který odstraňuje anomálie navrženého datového modelu. Je založena na postupné dekompozici datového modelu, kdy rozdělí atributy do většího počtu relací. Výsledkem normalizace je odstranění anomálií z modelu. Obecně je definováno šest normálních forem, které se značí 1NF, 2NF, 3NF, BCNF, 4NF a 5NF. V praxi se většinou používají jen první 3, a také se používá opačný proces – denormalizace. Ten slouží k tomu, že snižuje počet relací, a tím zvyšuje výkon databáze. (Šimonová a Panuš, 2007), (Novák, 2014)

První normální forma (1NF)

Definice první normální formy (1NF) souvisí s dodržением vlastností relací. Relace je v první normální formě, pokud neobsahuje vícehodnotové atributy. (Kaluža a Kalužová, 2012)

Druhá normální forma (2NF)

Definice této formy vychází z první normální formy a přidává další pravidlo. Relace je v druhé normální formě, je-li v 1NF a každý neklíčový atribut je plně funkčně závislý na celém primárním klíči. Z toho vyplývá, že pokud je primární klíč jen jedno pole, pak je automaticky relace v 2NF, jestliže je v 1NF. (Kaluža a Kalužová, 2012)

Třetí normální forma (3NF)

Tato normální forma opět souvisí s předchozí normální formou. „*Ve třetí normální formě je relace, jestliže je v 2NF a každý neklíčový atribut je netranzitivně závislý na primárním klíči.*“ (Kaluža a Kalužová, 2012, str. 81) Relace je tedy ve 3NF, pokud jsou všechny neklíčové atributy navzájem nezávislé.

Boyce Coddova normální forma (BCNF)

„*Relace je v Boyce-Coddově normální formě právě tehdy, jestliže každý determinant funkční závislosti v relaci je zároveň kandidátním klíčem.*“ (Kaluža a Kalužová, 2012, str. 81) Znamená to tedy, že mezi kandidátními klíči nesmí být žádná funkční závislost.

Vztah mezi BCNF a 3NF je takový, že každá relace, která je v BCNF je automaticky v 3NF. Naopak toto ovšem platit nemusí, protože může existovat situace, kdy nastane:

- v relaci je více kandidátních klíčů,
- všechny kandidátní klíče jsou složené,
- existuje atribut, který je společný pro všechny kandidátní klíče. (Kaluža a Kalužová, 2012), (Novák, 2014)

Čtvrtá normální forma (4NF), pátá normální forma (5NF)

„*Ve čtvrté normální formě (4NF) je relace tehdy, je-li v BCNF a všechny vícehodnotové závislosti obsažené v relaci jsou zároveň funkčními závislostmi.*“ (Kaluža a Kalužová, 2012, str. 82)

Relace je v 5NF, když je ve 4NF a tuto relaci již nelze bezztrátově rozložit.

2.3.2.3 Relační algebra

Relační model je postaven na matematických základech. V rámci relačního modelu jsou použité dva významné prostředky. Jsou to relační algebra a relační kalkul. Relační algebra je prostředek, který umožňuje práci s daty v tabulkách. Použitím operací relační algebry nad nějakou relací vrátí opět relaci. Operátory relační algebry se dělí na:

- tradiční operáty (sjednocení, průnik, rozdíl, součin),
- speciální operátory (selekce, projekce, spojení, dělení). (Šimonová a Panuš, 2007)

2.3.2.4 Návrh metodiky relačního modelování

Jedná se o proces, který transformuje konceptuální model v podobě ER diagramu do logického modelu. Tato transformace probíhá s ohledem na konečnou implementaci databáze a podle toho se volí databázový model. V tomto případě se jedná o relační databázový model.

1. **Vytvoření soustavy předběžných relací** – vychází z konceptuálního modelu. Předběžné relace jsou určeny na základě entit z návrhu ER diagramu konceptuálního modelu. Relacím se přiřadí pouze jména, kandidátní a cizí klíče. Tento krok abstrahuje od neklíčových atributů, se kterými prozatím nepracuje.
2. **Přiřazení zbývajících atributů** – v předchozím kroku byly vytvořeny předběžné relace. V tomto kroku se těmto relacím přiřadí ostatní atributy tak, aby byly závislé na celém primárním klíči.
3. **Normalizace modelu** – v tomto kroku probíhá proces normalizace nad všemi navrženými relacemi v předchozích krocích, které tvoří datový model. Proces má za úkol odstranit anomálie navrženého datového modelu. Uvádí se, že pro praktické účely stačí, aby navržené relace vyhovovaly BCNF.
4. **Revize konceptuálního modelu** – mělo by dojít k revizi modelu z důvodu úprav konceptuálního modelu, které vyvolávají vytvoření nových entit a vztahů.
5. **Specifikace domén** – toto je poslední krok návrhu metodiky relačního modelování. V tomto kroku jde o přiřazení charakteristik atributům, jako jsou datový typ, počet znaků, povolené hodnoty, struktura hodnot, kandidátní klíče či textový popis. (Kaluža a Kalužová, 2012)

2.3.3 Fyzický model

Třetím krokem datového modelování je vytvoření fyzického modelu. Tento model vychází z předchozích dvou částí – konceptuálního a logického modelu. Za vstup fyzickému modelu

jsou výstupy logického modelu. Jedná se realizaci vlastní databázové aplikace na platformě určitého softwaru. V této práci se jedná o Microsoft Access, který je popsán v další části práce.

2.4 SQL

Structured Query Language (SQL) je univerzální jazyk relačních databází, který je podporován většinou systémů řízení báze dat (SŘBD). Jazyk SQL vyvinula skupina výzkumníků koncem 70. let ve společnosti IBM. Původně byl tento jazyk pojmenován „*SEQUEL*“, ale zjistilo se, že tento název je již používán, a tak byl název jazyka zkrácen na „*SQL*“.

SQL je nejrozšířenější jazyk, který je používán pro komunikaci s relačními databázemi. Používá se zejména k jejich správě a údržbě, ale není příliš vhodný pro obecné programování databází. Funguje na principu, že zasílá požadavky - dotazy (query) na databázi. Databáze na základě tohoto požadavku odpoví. Jazyk SQL je řazen mezi deklarativní jazyky. Znamená to, že jazyk se dotazuje na výsledky dat z databáze. Nemusí se přitom definovat, jak tyto výsledky z databáze získat.

Příkazy jazyka SQL jsou děleny na několik kategorií. Jsou děleny v závislosti na své funkci:

- Data Definition Language (DDL),
- Data Query Language (DQL),
- Data Manipulation Language (DML),
- Data Control Language (DCL),
- příkazy řízení transakcí. (Oppel, 2008)

2.4.1 Data Definition Language (DDL)

DDL obsahuje příkazy, které souvisí s klíčovými slovy CREATE, ALTER a DROP. Příkazy DDL slouží k vytváření objektů (např. tabulek, indexů) a případnému měnění jejich struktury. Tyto příkazy mají vliv pouze na kontejnery, ve kterých jsou data v databázích uložena. Žádný z těchto příkazů neumožňuje vytvářet či upravovat data v tabulkách:

- CREATE – umožňuje vytvářet databáze, tabulky či indexy,
- ALTER – změna typ databázového objektu,
- DROP – smaže existující databázový objekt.

2.4.2 Data Query Language (DQL)

Příkazy v této kategorii souvisí s klíčovým slovem SELECT. DQL umožňuje vybírat data z databáze, která mohou být dále zobrazena nebo zpracována. Platí, že odpověď databáze na příkaz SELECT je ve tvaru tabulky. (Oppel, 2008)

2.4.3 Data Manipulation Language (DML)

Jsou to především příkazy INSERT, UPDATE a DELETE. Příkazy DML umožňují data do databáze přidávat, upravovat je či mazat. Všechny tyto příkazy mohou manipulovat s daty pouze jedné tabulky:

- INSERT – přidává do tabulky nové řádky,
- UPDATE – aktualizuje řádky v tabulce,
- DELETE – smaže řádky z tabulky. (Oppel, 2008)

2.4.4 Data Control Language (DCL)

Tyto příkazy souvisí se zabezpečením databáze. Mezi DCL příkazy patří především GRANT a REVOKE. Používají se především k udělování přístupu do databáze a nastavování oprávnění:

- GRANT – přiděluje oprávnění uživateli databáze,
- REVOKE – odebrá oprávnění uživateli databáze. (Oppel, 2008)

2.4.5 Příkazy řízení transakcí

Definice transakce: „*Transakce je oddělená sada akcí, které je nutné buď zpracovat jako celek, nebo vůbec.*“ (Oppel, 2008, str. 161) Existují 4 základní vlastnosti, které se skrývají pod akronymem ACID.

- **Atomicita (atomicity)** – transakce musí být provedena jako celek. Pokud je transakce úspěšně zakončená, pak musí být změny zapsány do databáze. Po každém úspěšném ukončení transakce se zavolá proces potvrzení (commit), který potvrdí změny a uloží je do databáze. V případě neúspěšného ukončení transakce, jsou všechny změny navráceny do původního stavu před započítáním transakce (rollback).
- **Konzistence (consistency)** – transakce převádí stav databáze z jednoho konzistentního stavu do druhého.
- **Izolace (isolation)** – transakce by měly být nezávislé jedna na druhé, i v případě, že se provádějí současně.

- **Trvanlivost (durability)** – změny, které byly provedeny transakcí, by měly být trvalé i při havárii nebo vypnutí databáze. (Oppel, 2008)

2.5 Microsoft Access 2010

Práce je orientována na aplikaci Microsoft Access 2010, která je součástí balíku programů Microsoft Office 2010. Je důležité brát zřetel na to, že společnost Microsoft aplikaci Access stále vyvíjí a inovuje. Znamená to, že v nových verzích (Microsoft Access 2013) mohou být odlišnosti v důsledku inovací aplikace.

Microsoft Access je aplikace sloužící jako osobní databázový systém, který kombinuje systém řízení báze dat s generátorem aplikací. Část systému řízení báze dat souvisí s vytvářením, správou či zpracováním dat v databázi, což jsou jeho základní funkce. Část generátoru aplikací souvisí s prvky, které umožňují vytvářet formuláře, dotazy, sestavy a další prvky, které souvisí s aplikací. Dotazy lze v programu vytvářet pomocí návrhového zobrazení, které generuje SQL dotazy. Navíc je v rámci programu Microsoft Access integrován programovací jazyk VBA (Visual Basic for Application). (Kroenke a Auer, 2015)

2.5.1 Relace (tabulky)

Jak již bylo popsáno v předchozím textu, relace jsou objekty, které se jeví uživatelům jako dvourozměrné tabulky. V aplikaci Microsoft Access se používá pro relaci pojem tabulka. V tabulkách jsou ukládána data. Každá tabulka je složena z n-tic, které představují záznamy, a atributů, které představují pole. Každá taková tabulka reprezentuje entitu navrženou v předchozích fázích datového modelování. V každé tabulce by mělo být obsaženo alespoň jedno pole, které by mělo představovat primární klíč. Primární klíč představuje jednoznačnou identifikaci záznamů v tabulce. (Hernandez, 2006)

ID_zakaznika	Jméno	Příjmení	Datum narození	Ulice	Číslo popisné	Město
as19	Alena	Asíková	14.6.1987	Nová	150	Rychnov
HA1	Jan	Hálek	78	Nová	456	Sviadnov
KR612	Martina	Křiváková	87	Křivá	124	Frýdek-Místek
MA4	Jana	Malá	90	Hnědá	412	Praha
MAL1	Petr	Malý	4.5.1978	Silniční	14	Brno
MY7	Jakub	Myšek	7.7.1965	Zelená	635	Staříč
NAV1		Navrátil	3.9.1987	Rovná	79	Sviadnov
NOV1	Lukáš	Novotný	12.8.1998	Nová	4165	Ostrava
PI8	Tomáš	Písař	8.8.1991	Hrad	96	Benešov
SK1	Marie	Školní	6.9.1974	U Lesa	18	Kocourkov
SO3	Marek	Socha	7.8.1974	Černá	145	Náchod
ZE8	Tereza	Zelená	7.7.1977	Fialová	146	Baška

Obrázek 2.1: Popis tabulky v aplikaci Microsoft Access (zdroj: vlastní)

Tabulky jsou děleny podle toho, co reprezentují:

- **objektové** – reprezentují něco hmatatelného (věc, osobu, místo),
- **událostní** – reprezentují událost (soudní slyšení, výsledky testů).

Další možností dělení tabulek je podle toho, jak se s daty uvnitř nich pracuje:

- **datová tabulka** – s daty v této tabulce se manipuluje (mění, přidávají, mažou),
- **validační tabulka** – data v takovéto tabulce jsou statická a slouží k implementaci integrity dat. (Hernandez, 2006)

Tabulky jsou základním objektem Microsoft Access. Pomocí tabulek je tvořena databáze. Aplikace Microsoft Access podporuje čtyři různé zobrazení tabulky.

- **Návrhové zobrazení** - slouží především k vytváření tabulek, v rámci kterého se definuje jejich struktura (zadávají se názvy, datové typy sloupců, primární klíče apod.).
- **Zobrazení datového listu** - v rámci tohoto zobrazení je možno data zobrazit, upravit či vkládat.
- **Zobrazení kontingenční tabulky** - slouží především k analýze dat v databázi.
- **Zobrazení kontingenčního grafu** – jedná se o grafickou podobu kontingenční tabulky.

Z těchto zobrazení jsou nejvíce používanými návrhové zobrazení a zobrazení datového listu. (Písek, 2011)

2.5.2 Základní datové typy

Datový typ představuje vlastnost, kterou sdílí všechny hodnoty obsažené v určitém poli. V aplikaci Microsoft Access je na výběr několik datových typů.

- **Text** – umožňuje zadávat textový řetězec (alfanumerické znaky) až do velikosti 255 znaků, je možno omezit délku tohoto řetězce.
- **Memo** – jedná se o textový řetězec bez omezení znaků, maximální velikost je 63999 znaků. Tento datový typ se používá zejména pro zadávání podrobných popisů.
- **Číslo** – umožňuje zadávat čísla.
- **Datum a čas** – pomocí tohoto datového typu se zadává datum a čas (datum narození).
- **Měna** – speciální datový typ pro zadávání měny.
- **Automatické číslo** – jedná se o číslo, které se automaticky generuje záznamy. Souvisí s primárním klíčem.

- **Ano/ne** – tento datový typ souvisí s logickou hodnotou, která značí ano nebo ne (true nebo false).
- **Hypertextový odkaz** – jedná se o textový řetězec, který odkazuje na jiné místo v síti.
- **Průvodce vyhledávání** – slouží k vytvoření pole se seznamem hodnot, které je buď možno propojit s polem jiné tabulky, nebo vytvořit ručně.
- **Příloha** – ukládají se zde například digitální fotografie. Datový typ příloha je podporován až od verze Microsoft Access 2007.
- **Objekt OLE** – lze ukládat binární data (obrázky, zvuk, dokument aplikace Microsoft Word), která jsou vytvořena v jiných aplikacích. (Písek, 2011), (JaknaOffice.cz, 2011)

2.5.3 Vlastnosti základních datových typů

Každý datový typ má své vlastnosti, které jsou předdefinované a lze je změnit. Výčet vlastností, které lze měnit, záleží na zvoleném datovém typu. Platí, že některé vlastnosti jsou společné pro více datových typů.

- **Velikost pole** – u datového typu text určuje jeho velikost. Výchozí hodnota je 255 znaků. Platí, že je v některých případech vhodné předdefinovat velikost pole (PSC). U datového typu číslo určuje rozsah zadávaného čísla.
- **Počet desetinných míst** – souvisí s datovými typy číslo a měna. Udává počet desetinných míst, které budou zobrazeny v poli.
- **Formát** – tato vlastnost určuje, v jakém formátu se budou data zobrazovat a případně i tisknout.
- **Vstupní maska** – pomocí zástupných znaků určuje zadávání dat do pole. Mezi zástupné znaky patří například: 0 pro povinné číslo, 9 pro nepovinné číslo, L pro povinné písmeno, ? pro nepovinné písmeno atd.
- **Titulek** – představuje nápis zobrazující se v záhlaví pole.
- **Výchozí hodnota** – představuje přednastavenou hodnotu, která se automaticky vkládá do pole.
- **Ověřovací pravidlo** – zadává se zde výraz, kterému musí vyhovovat data zadána do pole (například, aby nešlo zadat záporný věk).
- **Ověřovací text** – souvisí s ověřovacím pravidlem. Pokud zadaná data nevyhovují podmínce v ověřovacím pravidle, pak se zobrazí ověřovací text.
- **Je nutno zadat** – je-li hodnota nastavena na Ano, pak je nutno vyplnit toto pole.

- **Povolit nulovou délku** – vlastnost představuje, zda je možno prázdné (nevyplněné) pole brát jak platnou hodnotu pole.
- **Indexovat** - nastavení indexování pomáhá zrychlit vyhledávání a řazení záznamů. (Písek, 2011)

2.5.4 Vztahy

Aplikace Microsoft Access patří mezi relační databáze. Jak už bylo popsáno, tyto databáze jsou charakteristické tím, že ukládají data v relacích a mezi těmito relacemi definují vzájemné vztahy.

Aplikace MS Access umožňuje vytvořit 3 typy kardinality vztahů, které byly popsány v bodě 2.3.1.2:

1. vztah 1:1,
2. vztah 1:N,
3. vztah M:N – tento vztah je vytvořen pomocí dvou vztahů 1:N, které ukazují do pomocné relace složené pouze z primárních klíčů relací 1:N. (Písek, 2011)

Vztahy v aplikaci Microsoft Access se vytvářejí pomocí nástroje relace. Z karty databázové nástroje se vybere položka relace. Zde se vyberou požadované tabulky, mezi kterými mají být nadefinovány vztahy. Vazby se definují tak, že se vybere položka v jedné tabulce a přesměruje na položku z druhé tabulky, která musí mít stejný datový typ. Dále jsou zde na výběr možnosti:

- **Zajistit referenční integritu** – aplikace zajistí dodržování pravidel při práci s tabulkami, aby například nebylo možno smazat záznam v jedné tabulce, který souvisí s jinou tabulkou.
- **Aktualizace souvisejících polí v kaskádě** – tato možnost zajistí, že pokud dojde ke změně záznamu v jedné tabulce, tak se tato změna projeví i v souvisejících záznamech v dalších tabulkách.
- **Odstranění souvisejících polí v kaskádě** – tato možnost znamená, že pokud dojde ke smazání záznamu v jedné tabulce, tak se tento záznam smaže i v souvisejících tabulkách. (Písek, 2011), (Žárská, 2012)

2.5.5 Formuláře

Dalším objektem hojně používaným v aplikaci Microsoft Access je formulář. Jedná se o část aplikačního rozhraní, které slouží především k zobrazení a zadávání dat do databáze. Zadávání dat pomocí formulářů snižuje chybovost, protože se již uživatel nemusí starat o to, jestli

vyplňuje správné řádky apod. Další možností použití formulářů je jako přepínacích panelů sloužících k otevírání jiných formulářů. (Žárská, 2012)

Existuje několik zobrazení, pomocí kterých lze formuláře zobrazit a případně i upravovat. Mezi jednotlivými možnostmi zobrazení se lze libovolně přepínat.

- **Formulářové zobrazení** – jedná se o zobrazení, ve kterém pracuje standardní uživatel. V tomto zobrazení je možno data prohlížet či upravovat.
- **Návrhové zobrazení** – toto zobrazení slouží zejména k návrhu struktury formuláře, právě v tomto zobrazení lze formulář vytvořit či měnit. V tomto zobrazení nejsou dostupná zdrojová data.
- **Zobrazení rozložení** – funkčnost tohoto zobrazení je na pomezí dvou přechozích. Jednak se v jednotlivých polích zobrazují zdrojová data a zároveň je možno upravovat velikost ovládacích prvků.
- **Zobrazení datového listu** – formulář je zobrazen jako datový list. (Žárská, 2012), (Písek, 2011)

Vytvoření formuláře lze provést několika způsoby.

- **Automatický formulář** – jedná se o nejrychlejší způsob, jak vytvořit formulář. Problém této volby je v podobě omezení zdroje dat pouze z jedné tabulky.
- **Vytvoření formuláře pomocí průvodce** – umožňuje snadno vytvořit formulář v několika krocích.
- **Vytvoření formuláře ručně v návrhovém zobrazení** – jedná se o nejpracnější způsob, ale zároveň je možno formulář přizpůsobit, jak je třeba.

Zároveň lze do formulářů vkládat ovládací prvky. Existují dva druhy ovládacích prvků:

- **Vázané** – údaje, které zobrazují, mají vztah k datům v databázi.
- **Nevázané** – zobrazují data, která nemají vztah k datům v databázi. (Písek, 2011)

2.5.6 Tiskové sestavy

Další objektem aplikace Microsoft Access jsou sestavy. Sestavy jsou ideálním nástrojem, jak prezentovat data z databáze v tištěné podobě. Tiskové sestavy nijak neovlivňují data v tabulkách, pouze je zobrazují. (Písek, 2011)

Se sestavami lze v aplikaci Microsoft Access pracovat v několika režimech zobrazení, které slouží k editaci struktury sestavy nebo i zobrazení dat. Režimy zobrazení pro sestavy jsou obdobné těm pro formuláře.

- **Zobrazení sestavy** – v tomto zobrazení jsou data zobrazena a lze i upravovat uspořádání prvků v sestavě.
- **Návrhové zobrazení** – slouží k úpravě struktury sestavy. V tomto zobrazení se mohou sestavy vytvářet. Obdobně jako ve formulářích, nejsou v tomto zobrazení zobrazena data databáze.
- **Zobrazení rozložení** – je funkčně stejné jako u zobrazování formulářů. V sestavě jsou obsažena data a lze měnit rozmístění prvků sestavy.
- **Náhled** – zobrazí sestavu přesně tak, jak bude vytištěna na papír.

Možnosti, jak vytvořit sestavu, jsou obdobné těm, pro vytváření formulářů.

- **Automatická sestava** – sestava je vytvořena pomocí jednoho tlačítka.
- **Průvodce sestavou** – umožňuje vytvořit sestavu v několika krocích, ve kterých se například volí pole, která mají být v sestavě zobrazena apod.
- **Vytvoření sestavy v návrhovém zobrazení** – vytváření probíhá z prázdné sestavy, kdy je potřeba vše nastavit a upravit. (Písek, 2011)

2.5.7 Filtry

Když jsou data uložena v databázi, je potřeba tato data na základě určitých kritérií vybírat, aby se s nimi dalo dále pracovat a nebylo přitom nutné procházet všechny záznamy v databázi. Tento výběr lze v aplikaci Microsoft Access provést pomocí dvou dostupných prostředků. Tím prvním je použití filtrů. (Písek, 2011)

Jedná se o nejrychlejší a nejsnazší způsob výběru dat. Filtry fungují na principu, že pracují s daty, která vyhovují zadaným kritériím. Filtry se ukládají spolu s tabulkou jako její vlastnost. Aplikace Access pracuje s několika typy filtrů.

- **Filtr podle výběru** – jedná se o nejrychlejší typ filtrování, je založen na zobrazování záznamů, které mají společné rysy. Využívá se na příklad u tabulky, která má mnoho záznamů a je potřeba vyfiltrovat jen ty, které souvisí jen s určitým kritériem.
- **Filtr mimo výběr** – princip tohoto kritéria je založen na tom předchozím. Rozdíl je, že tento filtr záznamy související s daným kritériem skryje.
- **Filtr podle formuláře** – využívá se, když je třeba vybírat hodnoty ze seznamu nebo použít složitější podmínku jako kritérium.

- **Rozšířený filtr či řazení** – vychází z filtru podle formuláře, má rozšířené možnosti v podobě zadání způsobu řazení záznamů. Tvorba toho filtru je velice podobná tvorbě dotazů, které jsou podrobněji popsány dále. (Písek, 2011)

2.5.8 Dotazy

Druhou možností výběru dat z databáze je použití dotazů. Využití dotazů ale není jen pro výběr dat, dotazy se dají použít pro veškerou manipulaci s daty v databázi (mazání, aktualizace, filtrování). Dotazy mají oproti filtrům několik důležitých rozdílů.

- **Ukládání dotazů** – dotazy se ukládají jako samostatné objekty databáze, které jsou odděleny od ostatních objektů. Z toho vyplývá, že není problém uložený dotaz kdykoli znovu použít později.
- **Parametrický dotaz** – lze deklarovat dotaz, který se před vykonáním zeptá na zadání parametru, podle kterého má vykonat danou operaci.
- **Výběr dat z více tabulek** – aplikace umožňuje vybírat ze všech tabulek v rámci celé databáze.
- **Zadání dotazu v SQL** - pokud by dotaz nešlo navolit, je možno zadat dotaz v jazyce SQL. (Kruczek, 2011), (Písek, 2011)

V aplikaci Microsoft Access je definováno několik typů dotazů.

- **Výběrové dotazy** – jedná se o nejčastěji používané dotazy, jsou nejvíce podobné filtrům. Výhoda tohoto dotazu oproti filtrům je ta, že dotaz může vybírat data z několika tabulek.
- **Křížové dotazy** – umožňují zobrazit různé přehledy, kdy jsou jedny údaje závislé na druhých
- **Akční dotazy** – upravují údaje v databázi. Existují čtyři druhy akčních dotazů:
 - odstraňovací dotazy – vyberou záznamy na základě kritéria a vymažou je,
 - aktualizací dotazy – upravují data v databázi na základě kritéria,
 - přidávací dotazy – přidají záznam do tabulky,
 - vytvářecí dotazy – uloží výsledek dotazu do nově vytvořené tabulky. (Písek, 2011)

2.5.9 Makra

Mezi další objekty aplikace Microsoft Access patří makra. Makro je posloupnost akcí, které jdou za sebou a vykonávají se.

Výhodou maker je jejich snadná tvorba a použití, což vyplývá z možnosti tvořit makra v českém jazyce, pokud je aplikace Microsoft Access v české verzi. Makra jsou někdy až příliš jednoduchá, a proto se nehodí k tvorbě všeho, co by bylo potřeba. Toto částečně řeší jazyk VBA (Visual Basic for Application), který umožňuje psát makra. Nevýhodou tohoto je nutná znalost jazyka VBA. (Kruczek, 2011)

2.5.10 Visual Basic for Application (VBA)

V rámci balíku programů Microsoft Office (Access, Excel, Word) je integrován programovací jazyk Visual Basic for Application (VBA). Jedná se o programovací jazyk sloužící tvorbě maker, které rozšiřují funkce balíku programu Office nebo slouží k tvorbě nových aplikací.

Visual Basic for Application je interpretovaný objektově orientovaný programovací jazyk. Interpretovaný jazyk znamená, že napsaný kód je při spuštění řádek po řádku přeložen do strojového kódu a okamžitě vykonán. Aplikace Microsoft Access je složena z objektů (načtená databáze, tabulky, formuláře, atd.), všechny tyto objekty mají své vlastnosti, události a metody, se kterými jde pomocí VBA programově manipulovat. (Novák, 2014), (Shepherd, 2012)

Pro psaní programů v jazyce VBA se používá speciální prostředí integrované v rámci aplikace Microsoft Access zvané Visual Basic Editor (VBE), ten je dostupný pod klávesovou zkratkou ALT+F11. Zdrojový kód je v rámci VBE psán do kontejnerů, kterým se říká moduly.

Moduly obsahují procedury, ze kterých se skládá zdrojový kód. Procedury jsou děleny na podrutiny a funkce. Podrutina je část zdrojového kódu, která přímo nevrací žádnou hodnotu. Naopak funkce je část zdrojového kódu, která vrací nějakou hodnotu. Moduly jsou děleny do dvou typů:

- standardní moduly – zde je psán takový zdrojový kód, který má být přístupný ze všech míst aplikace (univerzální funkce a podrutiny), aby se dal využívat v celé aplikaci. K tomu, aby byl vidět, musí být zdrojový kód veřejný,
- moduly třídy – souvisí s formuláři či sestavami. (Shepherd, 2012), (Novák, 2014)

3 Analýza současného stavu ve firmě

Databázová aplikace je vytvářena pro projekční kancelář DK projekt s.r.o. Tato firma byla založena v roce 2006 a sídlí v Ostravě.

Projekční kancelář má široké portfolio služeb, které zpracovává. Mezi tyto služby patří například projekční práce, kde zpracovává mimo jiné projekty vody, kanalizace, plynu či ústředního topení. Dále se zabývá inženýrskou činností, koordinací BOZP. Zpracovává průkazy energetické náročnosti budov, připravuje podklady pro program Nová zelená úsporám. Projekční kancelář dále působí jako Energetické konzultační a informační středisko.

V současné době jsou data o zákaznících projekční kanceláře a jejich projektech zpracovávána a uchovávána pomocí programů Microsoft Excel a Microsoft Word. Veškerou práci s těmito daty provádí jeden uživatel na jednom počítači. Pomocí tohoto řešení jsou některé úkony s daty, jako je jejich správa či vyhledávání v nich, zbytečně komplikované a zdlouhavé. Hodilo by se tedy některé úkony zautomatizovat a některé nedostatky odstranit. Toto by měla vyřešit jednotná databázová aplikace, která bude sloužit projekční kanceláři jako jednotný systém evidence, aby tedy nebylo potřebné používat jiných aplikací.

3.1 Požadavky na aplikaci

Databázová aplikace by měla zahrnovat tabulky, do kterých bude možno ukládat informace důležité pro projekční kancelář. Nejdůležitější je zaznamenávat informace o projektech, které projekční kancelář zpracovává. Projekty by se měly dát dělit na ty, které jsou již hotové a ty, které hotové nejsou. Všechny projekty jsou zpracovávány pro zákazníky, ti jsou jak soukromé osoby, tak firmy a je nutno je rozlišit. Dále by databázová aplikace měla zachycovat informace o službách, které firma nabízí. Také by měli být v databázi zachovány informace o zpracovatelích, kteří zpracovávají jednotlivé projekty.

Požadovaná databázová aplikace by měla zvládat veškeré úkony s daty v tabulkách databáze (vkládání, zobrazování, úprava či mazání) tak, aby mohly být prováděny i běžným uživatelem, který nemá znalosti SQL dotazů a podobně. Dále by mělo být možné jednoduše sestavovat nejrůznější výpisy o datech z databáze pro tisk. Samozřejmostí je, aby bylo ovládání výsledné aplikace intuitivní a přehledné, a tedy přechod ze stávajícího systému co nejsnazší.

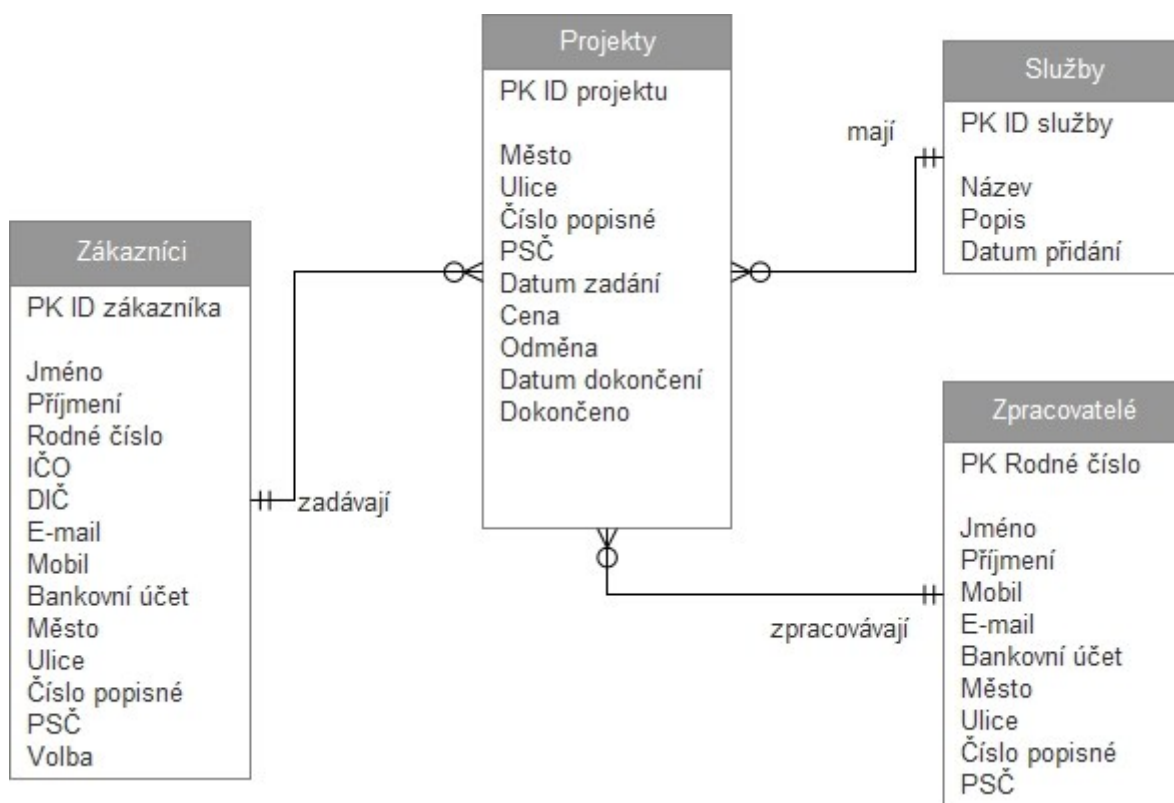
4 Návrh a implementace databázové aplikace

Postup návrhu databázové aplikace je rozčleněn do třech kroků. Tyto kroky již byly popsány v teoretické části v rámci části datového modelování.

Prvním krokem je vytvoření konceptuálního modelu, který popisuje návrh systému v ER digramu. Druhým krokem je logický návrh, který převádí konceptuální model do některého databázového modelu. V rámci této práce se jedná o relační databázový model. Třetím krokem je fyzický model, který implementuje navržený logický model na určitou platformu softwaru. V rámci této práce se jedná o převedení logického modelu do prostředí Microsoft Access.

4.1 Konceptuální model

Konceptuální model vytváří obecný pohled na databázi. Popisuje obecně pomocí entit, jak jsou data uložena v databázi a určuje vztahy mezi nimi. V rámci entit jsou atributy, které popisují jejich vlastnosti.



Obrázek 4.1: Zobrazení návrhu konceptuálního modelu pomocí ER diagramu

4.1.1 Popis entit

V rámci konceptuálního modelu byly vytvořené čtyři entity (Obrázek 4.1). Tyto entity obecně popisují nejdůležitější prvky databáze a jejich vlastnosti v rámci atributů. Mezi entitami jsou naznačeny vazby, jak by měly být propojeny.

Zákazníci

Tato entita představuje obecně zákazníka, který zadává projekty. Firma si o zákaznících ukládá informace, které reprezentují atributy entity. Zákazník může být jak soukromá osoba, tak firma. Z tohoto důvodu jsou v entitě obsaženy vlastnosti pro obě tyto osoby.

Primárním klíčem by v této entitě mělo automaticky inkrementované číslo (ID zákazníka). Vazba entity Zákazníci je spojena s entitou Projekty 1:N. Znamená to, že jeden zákazník může zadat více projektů.

Atributy entity Zákazníci a jejich stručný popis:

- ID zákazníka – primární klíč,
- jméno – představuje jméno soukromé osoby nebo název firmy,
- příjmení – příjmení soukromé osoby,
- rodné číslo – každá soukromá osoba má své vlastní rodné číslo,
- IČO (identifikační číslo osoby) – reprezentuje IČO přidělené firmě,
- DIČ (daňové identifikační číslo) – DIČ firmy, pokud má přiděleno,
- e-mail – e-mailová adresa zákazníka,
- mobil – telefonní kontakt na zákazníka,
- bankovní účet – bankovní spojení se zákazníkem,
- město – název města, kde zákazník bydlí,
- ulice – představuje název ulice zákazníkova bydliště,
- číslo popisné,
- PSČ (poštovní směrovací číslo) – každé město má své PSČ,
- volba – reprezentuje volbu, jestli se jedná o zákazníka jako soukromou osobu nebo firmu.

Zpracovatelé

Druhá popisovaná entita reprezentuje zpracovatele, který zpracovává projekty zadané zákazníkem. Firma si o zpracovatelích uchovává určité informace, které opět představují atributy entity.

Primární klíč v rámci entity Zpracovatelé by mělo být rodné číslo každého zpracovatele. Entita má vazbu 1:N na entitu Projekty. Znamená to, že jeden zpracovatel může zpracovávat několik projektů.

Atributy entity Zpracovatelé a jejich stručný popis:

- rodné číslo – primární klíč – každý zpracovatel je osoba, která má své rodné číslo,
- jméno – jméno zpracovatele,
- příjmení – příjmení zpracovatele,
- mobil – telefonní kontakt na zpracovatele,
- e-mail – e-mailová adresa,
- bankovní účet – bankovní spojení se zpracovatelem,
- město – název města bydliště zákazníka,
- ulice – název ulice bydliště,
- číslo popisné,
- PSČ (poštovní směrovací číslo).

Služby

Třetí entita reprezentuje služby, které firma nabízí zákazníkům a v rámci projektů je zpracovává.

Primárním klíčem entity Služby by mělo být automaticky inkrementované číslo (ID služby). Entita má vazbu na entitu Projekty 1:N. Z toho vyplývá, že jedna služba může mít několik projektů.

Atributy entity Služby a jejich stručný popis:

- ID služby – primární klíč,
- název – název služby,
- popis – širší popis služby,
- datum přidání – datum, kdy byla služba přidána mezi poskytované služby.

Projekty

Čtvrtá entita představuje projekty, které firma zpracovává. Tato entita propojuje všechny ostatní entity. Všechny projekty mají své vlastnosti, které jsou uchovány v rámci atributů entity.

Primární klíč této entity by mělo být automaticky inkrementované číslo (ID projektu). Na entitu Projekty mají všechny ostatní tři entity vazbou 1:N, kde je entita Projekty na straně N.

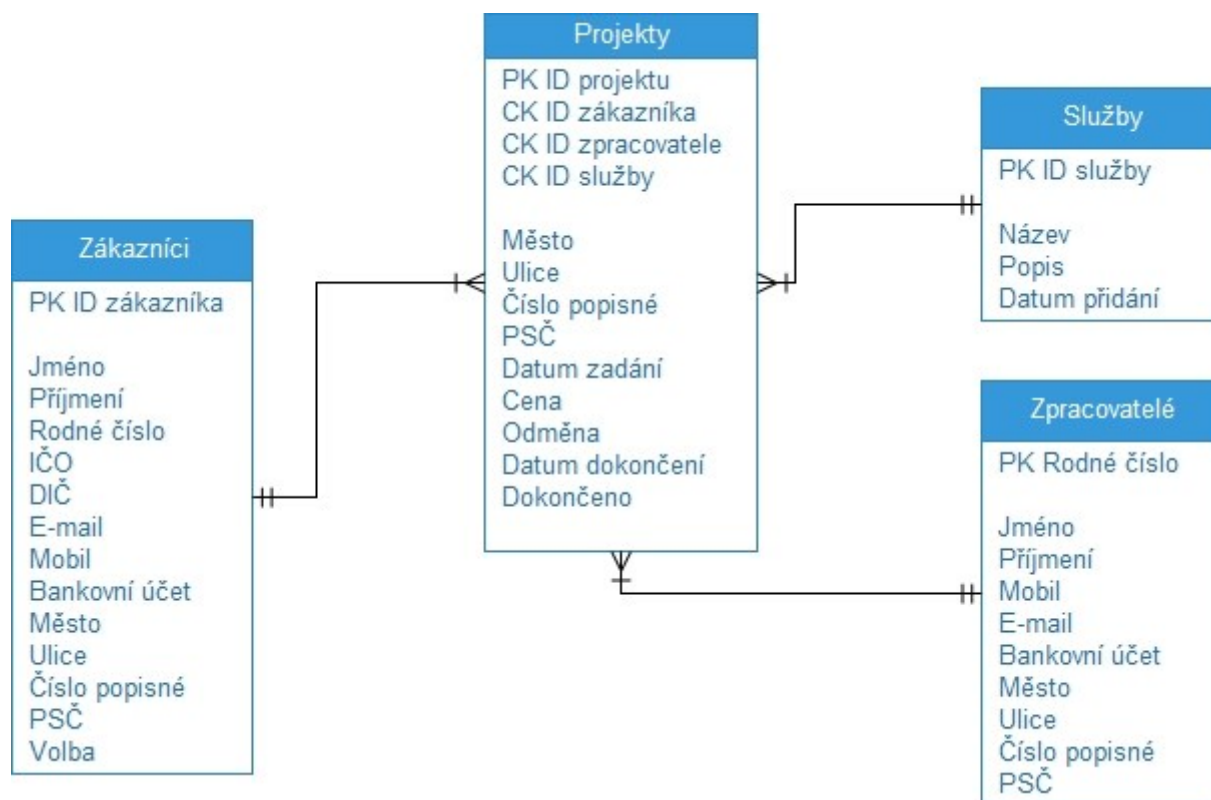
Atributy entity Projekty a jejich stručný popis:

- ID projektu – primární klíč,
- město – název města nemovitosti, na které se provádí projekt,
- ulice – název ulice nemovitosti, na které se provádí projekt,

- číslo popisné,
- PSČ (poštovní směrovací číslo),
- datum zadání – datum, kdy byl projekt zadán,
- cena – cena projektu,
- odměna – odměna pro zadavatele projektu,
- datum dokončení – datum, kdy byl projekt dokončen,
- dokončeno – označuje, zda je daný projekt dokončený.

4.2 Logický model

Návrh logického modelu vychází z konceptuálního návrhu a transformuje tento návrh do relačního databázového modelu, aby se tento návrh dal použít pro implementaci do aplikace Microsoft Access.



Obrázek 4.2: Zobrazení návrhu logického modelu pomocí ER diagramu

Obrázek 4.2 zobrazuje grafický návrh logického modelu. V rámci logického modelu byly entity z konceptuálního návrhu transformovány do relací (tabulek) tak, že pro každou entitu byla vytvořena samostatná relace (Zákazníci, Zpracovatelé, Projekty, Služby).

Každá relace má primární klíč. Všechny vztahy mezi entitami jsou 1:N. Do relací se tato vazba promítne tak, že v relaci Projekty se vytvoří cizí klíče, které odpovídají primárním klíčům z ostatních relací (CK ID zákazníka, CK ID zpracovatele, CK ID služby).

4.3 Aplikace Microsoft Access

Po vytvoření konceptuálního a logického návrhu přichází na řadu fyzická implementace databázové aplikace. Vytvořením předešlých modelů proběhl návrh databázové aplikace. Dalším krokem je převedení těchto navržených modelů do prostředí aplikace Microsoft Access. Implementace aplikace Microsoft Access vychází především z návrhu logického modelu.

Relace z logického návrhu jsou převedeny do tabulek, mezi těmito jsou vytvořeny vztahy. Nad těmito tabulkami je v aplikaci vytvořeno aplikační prostředí pomocí formulářů, sestav či dotazů.

4.3.1 Tabulky

Na základě relací z logického návrhu modelu (Obrázek 4.2) byly vytvořeny čtyři tabulky. Jedná se o tabulky, které jsou dále podrobně popsány:

- tblZakaznici,
- tblZpracovatele,
- tblSluzby,
- tblProjekty.

Tabulka tblZakaznici

Tabulka si uchovává informace týkající se všech zákazníků projekční kanceláře. Jsou to právě zákazníci, pro které firma zpracovává projekty. Primární klíč tabulky je pole *ID*, datový typ tohoto pole je automatické číslo, které se automaticky generuje s každým novým záznamem do tabulky a je v rámci ní unikátní, dále je u tohoto pole zajištěna vlastnost indexace bez duplicity. Tabulka je právě pomocí primárního klíče propojená s tabulkou tblProjekty.

Pole *Volba* je datového typu číslo a vyplňuje se na základě toho, jestli je zákazník soukromá osoba nebo firma. V tabulce jsou obsažena pole obsahující identitu uživatele – *Jmeno*, *Prijmeni*, *RodneCislo*, *ICO* a *DIC*. Pole *Jmeno* je nutno vždy zadat. Platí, že pole *Prijmeni* a *RodneCislo* se vyplňují jen pro zákazníka soukromou osobu – tedy když zákazník není firma. Pokud je zákazník firma, pak se vyplňují pole *ICO* a *DIC*. Platí, že v každém záznamu budou prázdná některá pole – tedy buď *Prijmeni* a *RodneCislo* nebo *ICO* a *DIC*. Další pole obsažená v tabulce

souvisí s kontaktováním zákazníka. Jsou to pole *Email*, *Mobil* a *BankovniUcet*. Poslední skupinou polí jsou pole související s adresou zákazníka. Patří sem pole *Mesto*, *Ulice*, *CisloPopisne* a *PSC*.

Tabulka 4.1: popis polí tabulky *tblZakaznici*

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska
ID	Automatické číslo	Dlouhé celé číslo	-	
Volba	Číslo	Celé číslo	Ano	
Jmeno	Text	60	Ano	
Prijmeni	Text	30	Ne	
RodneCislo	Text	10	Ne	000000\0000
ICO	Text	8	Ne	00000000
DIC	Text	12	Ne	LL0000000000
Email	Text	60	Ano	
Mobil	Text	9	Ne	" +420 "000\ 000\ 000
BankovniUcet	Text	20	Ne	000000\ - 00000000000\0000
Mesto	Text	30	Ano	
Ulice	Text	30	Ano	
CisloPopisne	Text	10	Ano	
PSC	Text	5	Ano	000\ 00

Tabulka *tblZpracovatele*

Tato tabulka uchovává informace týkající se zpracovatelů projektů. Jsou to zaměstnanci firmy. Primárním klíčem této tabulky je pole *RodneCislo*, které se zadává podle rodného čísla zaměstnance. Pole je datového typu text, má velikost 10 znaků, nutno zadat, je indexováno bez duplicity a vstupní maska tohoto pole je „000000\0000“. Tabulka *tblZpracovatele* je propojena s tabulkou *tblProjekty* pomocí primárního klíče.

V tabulce jsou obsažená pole, která obsahují základní identitu zpracovatele. Jsou to pole *RodneCislo*, *Jmeno*, *Prijmeni*. Dále jsou zde pole, která obsahují data o kontaktech zpracovatelů. Jsou to pole *Mobil*, *Email* či *BankovniUcet*. Poslední část polí obsažených v tabulce souvisí s adresou zpracovatele. Jsou to pole *Mesto*, *Ulice*, *CisloPopisne* a *PSC*.

Tabulka 4.2: popis polí tabulky tblZpracovatele

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska
RodneCislo	Text	10	Ano	000000\0000
Jmeno	Text	30	Ano	
Prijmeni	Text	30	Ano	
Mobil	Text	9	Ne	" +420 "000\ 000\ 000
Email	Text	60	Ne	
BankovniUcet	Text	20	Ne	000000\ - 00000000000\0000
Mesto	Text	30	Ano	
Ulice	Text	30	Ano	
CisloPopisne	Text	10	Ano	
PSC	Text	5	Ano	000\ 00

Tabulka tblSluzby

V této tabulce jsou ukládána data týkající se služeb, které firma nabízí a v rámci projektů je pro své zákazníky zpracovává. Primární klíč tabulky je automatické číslo, které se generuje s každým novým záznamem do tabulky, indexace pole je bez duplicity. Tabulka je propojená stejně jako ty předešlé s tabulkou tblProjekty pomocí primárního klíče.

Pole obsažená v tabulce jsou *Nazev*, *Popis* a *DatumPridani*. Při vkládání záznamu do tabulky je pole *Nazev* povinné, pole *Popis* je volitelné. Pole *DatumPridani* zaznamenává datum, od kdy byla daná služba přidána do portfolia služeb firmy.

Tabulka 4.3: popis polí tabulky tblSluzby

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska
ID	Automatické číslo	Dlouhé celé číslo	-	
Nazev	Text	60	Ano	
Popis	Text	255	Ne	
DatumPridani	Datum a čas		Ne	

Tabulka tblProjekty

Následuje popis klíčové tabulky tblProjekty. V této tabulce se schraňují data týkající se všech projektů, které firma zpracovává nebo již zpracovala. Primárním klíčem této tabulky je

pole *ID*, jedná se o automatické číslo, které zajišťuje jedinečnost záznamů. Toto pole je automaticky generováno s každým novým záznamem do tabulky a je indexováno bez duplicity.

Na tabulku *tblProjekty* mají všechny tři ostatní tabulky (*tblZakaznici*, *tblZpracovatele*, *tblSluzby*) vazby. Tyto vazby jsou vytvořeny pomocí cizích klíčů (*ID_sluzby*, *ID_zakaznika*, *ID_zpracovatele*), které odkazují na primární klíče ostatních tabulek. Díky těmto vazbám je možno při zadávání záznamu do tabulky vybírat hodnoty z těchto tabulek. To znamená, že při zadávání nového záznamu do tabulky se dá lehce vybrat, pro kterého zákazníka bude projekt zpracováván, jaká služba se zpracovává, a který zpracovatel daný projekt zpracovává.

Dále jsou v rámci této tabulky pole, která souvisí s adresou nemovitosti, na které se zpracovává daný projekt. Mezi tato pole patří *Mesto*, *Ulice*, *CisloPopisne* a *PSC*, nutno dodat, že všechna tato pole jsou povinná. Další skupina polí souvisí s datem, kdy byl projekt zadán a dokončen a samozřejmě také s jeho finanční stránkou. Jsou to pole *DatumZadani*, *Cena*, *Odmena*, *DatumDokonceni*. Speciální pole sloužící k indikaci, zda je daný projekt dokončen je pole *Dokonceno*. Toto pole má pouze dva možné stavy – ano/ne, kdy v případě ano je projekt dokončen a v případě ne, ještě projekt dokončen není.

Tabulka 4.4: popis polí tabulky tblProjekty

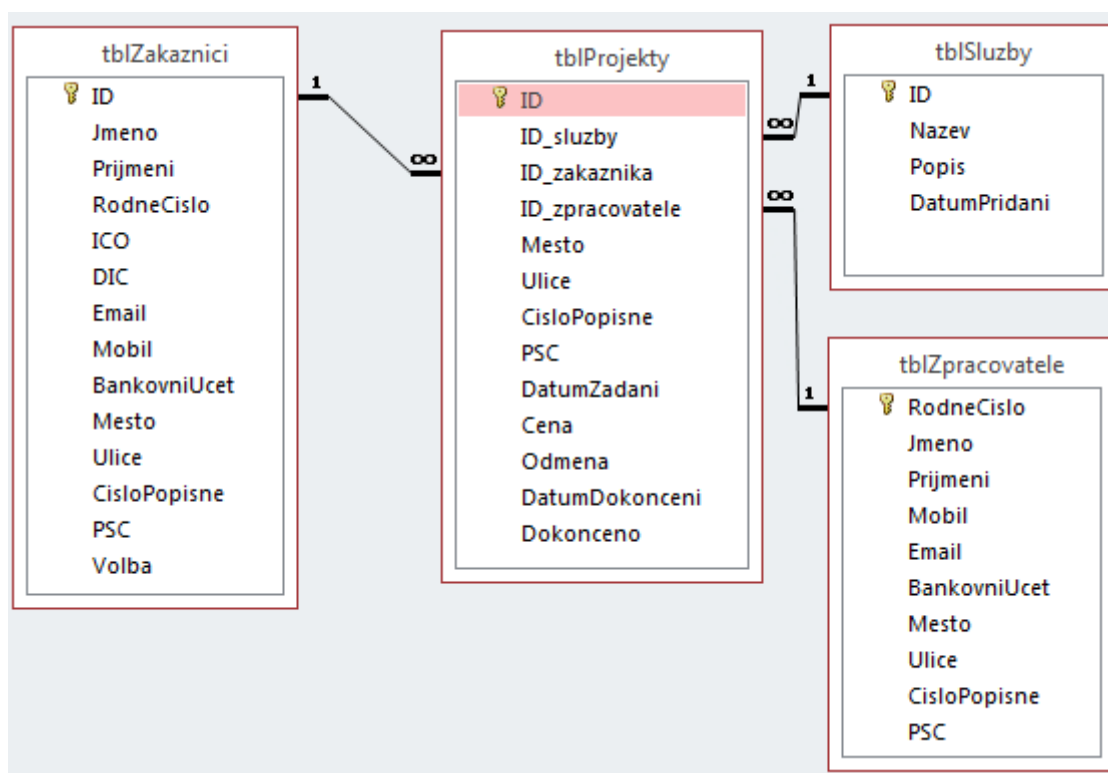
Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska
ID	Automatické číslo	Dlouhé celé číslo	-	
ID_sluzby	Číslo	Dlouhé celé číslo	Ano	
ID_zakaznika	Číslo	Dlouhé celé číslo	Ano	
ID_zpracovatele	Text	10	Ano	
Město	Text	30	Ano	
Ulice	Text	30	Ano	
CisloPopisne	Text	10	Ano	
PSC	Text	5	Ano	000\ 00
DatumZadani	Datum a čas		Ano	
Cena	Měna		Ne	
Odmena	Měna		Ne	

DatumDokonceni	Datum a čas		Ne	
Dokonceno	Ano/ne		Ne	

4.3.2 Vztahy

Obrázek 4.3 zobrazuje návrh tabulek a vztahů mezi nimi. Vztahy mezi tabulkami byly vytvořeny pomocí cizích klíčů, které jsou obsaženy v tabulce tblProjekty (ID_sluzby, ID_zakaznika, ID_zpracovatele) a odkazují vždy na příslušné primární klíče přidružených tabulek.

Všechny vztahy mezi tabulkami jsou 1:N. Znamená to, že jeden zákazník může mít zadáno několik projektů, jeden zpracovatel zpracovává více projektů a jedna služba může mít více projektů. Všechny vztahy mají vlastnosti, které jsou zajištění referenční integrity a kaskádové aktualizace souvisejících polí. Referenční integrita znamená, že aplikace Microsoft Access bude hlídat, aby nedošlo například k přidání záznamu do tabulky tblProjekty zákazníkovi, který neexistuje v tabulce tblZakaznici. Vlastnost kaskádové aktualizace souvisejících polí zajišťuje, že například změna provedená v tabulce tblZakaznici se projeví i v přidružené tabulce tblProjekty.



Obrázek 4.3: Návrh tabulek a vztahů

4.3.3 Formuláře

Následuje popis formulářů. Jsou to právě formuláře, pomocí kterých může běžný uživatel obsluhovat tabulky a celou aplikaci. Není tedy nutné, aby uživatel znal principy tvorby databází či SQL jazyk k ovládání databázové aplikace.

V rámci aplikace bylo vytvořeno mnoho formulářů sloužících ke správě dat v tabulkách (přidávání záznamů, úprava či jejich mazání) a samozřejmě formuláře sloužící k navigaci v aplikaci. Následující text se zmiňuje o těch formulářích, které jsou při práci s aplikací nejdůležitější. Platí, že mezi ty nejdůležitější formuláře patří především ty, které odpovídají tabulkám v aplikaci a pracují s jejich daty. Všechny hlavní formuláře byly vytvořeny v jednotném formátu s použitím stejných barev apod., aby se uživateli s aplikací příjemně a intuitivně pracovalo.

Formulář frmUvod

Jedná se o úvodní formulář, který se zobrazí po spuštění aplikace (Obrázek 4.4). Tento formulář nezobrazuje data tabulek, ale slouží k navigaci v aplikaci. V horní části formuláře je zobrazeno logo firmy. Dále jsou na formuláři rozmístěná tlačítka, která slouží k otevírání jiných formulářů (Zákazníci, Projekty, Nedokončené projekty, Služby a Zpracovatelé), které již s daty v tabulkách pracují. Ve spodní části formuláře je tlačítko Tisk, které zobrazuje formulář k tisknutí sestav. Pod tímto tlačítkem je červeně zvýrazněné tlačítko určené k zavření aplikace.



Obrázek 4.4: Formulářové zobrazení úvodního formuláře aplikace

Formulář frmZakaznici

Dalším popisovaným formulářem je *frmZakaznici* (Obrázek 4.5). Jedná se o formulář určený primárně k práci s tabulkou *tblZakaznici*. Znamená to, že data této tabulky jsou v tomto formuláři zobrazována. Pomocí tlačítek lze záznamy do tabulky přidávat, upravovat či je mazat, dále jsou zde dostupné tlačítka pro tisk sestav o zákaznících.

The screenshot displays the **Zákazníci** (Customers) form. At the top, there is a header bar with the title "Zákazníci" and a button "Nový zákazník" (New customer). Below the header, the form is divided into two main sections. The left section contains input fields for customer details: ID (9), Jméno (Petr), Příjmení (Kachlička), Rodné číslo (451214/1740), IČO, DIČ, E-mail (kachlicka@gmail.com), Mobil (+420 737 514 145), Bankovní účet (000000-0011241414/2014), Město (ostrava), Ulice (Nová), Číslo popisné (147/74), and PSČ (724 15). To the right of these fields are buttons for "Upravit zákazníka" (Edit customer), "Smazat zákazníka" (Delete customer), "Hledat" (Search), "Vyhledat" (Find), and "Zobrazit vše" (Show all). Below the input fields are navigation buttons: a red "ZPĚT" (Back) button, and four blue buttons for navigating between records. The right section contains buttons for "Tisknout celkový výpis zákazníka" (Print total customer list), "Tisknout seznam zákazníků" (Print customer list), and "Tisknout zákazníka a nedokončené projekty" (Print customer and incomplete projects). At the bottom, there is a table titled "Projekty:" (Projects:). The table has columns: ID, Název, Město, Ulice, Číslo popisné, PSČ, Zadáno, Cena, Odměna, and Dokončeno. It contains two rows of data: ID 6, Název Elektřina, Město Paskov, Ulice Nová, Číslo popisné 12, PSČ 718 45, Zadáno 26.2.2015, Cena 15 800,00 Kč, Odměna, Dokončeno; and ID 8, Název Elektřina, Město Ostrava, Ulice Nová, Číslo popisné 148, PSČ 718 45, Zadáno 26.2.2015, Cena 1 700,00 Kč, Odměna, Dokončeno 26.2.2015.

ID	Název	Město	Ulice	Číslo popisné	PSČ	Zadáno	Cena	Odměna	Dokončeno
6	Elektřina	Paskov	Nová	12	718 45	26.2.2015	15 800,00 Kč		
8	Elektřina	Ostrava	Nová	148	718 45	26.2.2015	1 700,00 Kč		26.2.2015

Obrázek 4.5: Formulářové zobrazení formuláře *frmZakaznici*

V záhlaví formuláře je tlačítko pro přidávání zákazníka. Po kliknutí na toto tlačítko se otevře formulář (Obrázek 4.6), přes který se přidává nový záznam do tabulky *tblZakaznici*. V tomto formuláři je nejdříve nutno zvolit právní formu (soukromá osoba, firma). Podle této volby se znepřístupní ta textová pole, která s příslušným zákazníkem nesouvisí. Po vyplnění textových polí se pomocí tlačítka přidat přidá záznam. Některá pole jsou povinná, v případě jejich nevyplnění se zahlásí chyba, která upozorňuje na daný nedostatek, a případně i aktivuje kurzor do příslušného pole. Tlačítko zavřít na tomto formuláři slouží k zavření formuláře.

Hlavní část formuláře slouží především k procházení jednotlivých záznamů z tabulky *tblZakaznici*. Tyto záznamy jsou zobrazeny na formuláři pomocí uzamčených textových polí. Ta jsou uzamčena z důvodu, aby nebylo možno provádět úpravy záznamů v tabulce pomocí tohoto formuláře a tím nedocházelo k nahodilým chybám. Pod těmito poli se nachází navigační

tlačítka pro přecházení mezi záznamy. Ve spodní části formuláře se nachází podformulář *frmPodformularProjekty*. Tento zobrazuje projekty zadané právě vybraným zákazníkem a informace o nich.

V pravé části formuláře se nachází textové pole sloužící k vyhledávání záznamů v tabulce *tblZakaznici*. To funguje na principu, že po kliknutí na tlačítko *Vyhledat* se vyhledají odpovídající záznamy v tabulce *tblZakaznici* s textem zadáním do textového pole a tyto záznamy jsou odfiltrovány. V případě nenalezení záznamu je zobrazena chybná hláška.

Dále je na formuláři tlačítko sloužící k úpravě zákazníka. Ten otevírá formulář *frmUpravitZakaznika* (Obrázek 4.6). Tento formulář funguje na podobném principu jako přidání zákazníka s tou výjimkou, že umožňuje upravit právě zobrazeného zákazníka ve formuláři. Není možno upravovat hodnotu pole ID, protože se jedná o primární klíč tabulky. Opět je v případě špatného vyplnění zobrazena chybová hláška a kurzor aktivován do textového pole. Tlačítko *Smazat zákazníka* slouží ke smazání záznamu právě zobrazeného zákazníka z tabulky *tblZakaznici*. Před smazáním je ještě zobrazeno dialogové okno, které žádá o potvrzení smazání záznamu.

The image shows two side-by-side windows from a software application. The left window is titled 'Upravit zákazníka' (Edit Customer) and the right window is titled 'Nový zákazník' (New Customer). Both windows have a similar layout. At the top, there is a section for 'Právní forma' (Legal form) with two radio buttons: 'Soukromá osoba' (Private person) and 'Firma' (Company). Below this, there are several text input fields for customer data. In the 'Upravit' window, the 'ID' field is pre-filled with the number '7'. The other fields contain data for a customer named 'Jan Malý' with address 'Ostrava, Zelená' and contact information. At the bottom of the 'Upravit' window are two buttons: 'Změnit' (Change) and 'Zavřít' (Close). The 'Nový' window has empty input fields for the same data. At the bottom of the 'Nový' window are two buttons: 'Přidat' (Add) and 'Zavřít' (Close).

Obrázek 4.6: Formuláře pro úpravu a přidání záznamu zákazníka

Nakonec jsou na formuláři umístěna tlačítka sloužící k tisku sestav o zákazníkovi. Tlačítko *Tisknout celkový výpis zákazníka* zobrazí sestavu, na které je kompletní výpis informací

zákazníka spolu se všemi projekty, které zadal. Oproti tomu tlačítko *Tisknout zákazníka a nedokončené projekty* zobrazí sestavu, na které je kompletní výpis zákazníka a pouze ty projekty, které nejsou dokončené. Posledním tlačítkem je *Tisknout seznam zákazníků*, to zobrazí seznam všech zákazníků projekční kanceláře.

Formulář frmProjekty

V aplikaci jsou vytvořeny dva velmi podobné formuláře sloužící k práci se záznamy tabulky *tblProjekty*. Prvním je formulář *frmProjekty* a druhým *frmVsechnyProjekty*. Jedná se o víceméně totožné formuláře. Rozdíl mezi nimi je ten, že formulář *frmVsechnyProjekty* pracuje se záznamy tabulky *tblProjekty* všech projektů a formulář *frmProjekty* pracuje pouze se záznamy těch projektů, které nejsou označeny jako dokončené.

Z výše uvedeného důvodu je v další části popisován formulář *frmProjekty* (Obrázek 4.7). Jak již bylo popsáno výše, tento formulář slouží k zobrazování záznamů tabulky *tblProjekty*. Pomocí tohoto formuláře je dále možno do tabulky přidávat nové záznamy, upravovat je nebo je mazat. Další možností je označení projektu jako dokončený a tisk sestav týkajících se daného projektu.

Nedokončené projekty

Nový projekt

ID:

11

Upravit projekt

ID služby:

4

Dokončit projekt

ID zákazníka:

8

Smazat projekt

ID zpracovatele:

8564123114

Město:

Ostrava

Ulice:

Rychlá

Číslo popisné:

98

PSČ:

719 84

Datum zadání:

3.3.2015

Cena:

14 000,00 Kč

Tisknout projekt

Odměna:

300,00 Kč

Tisknout nedokončené projekty

Hledat ID

Vyhledat

Zobrazit vše

Tisknout seznam projektů

◀

⏪

⏩

▶

Zákazník

Služba

Zpracovatel

ID	Jméno	Příjmení	Rodné číslo	IČO	Email	Mobil
8	Stavebnictví s.r.o.			14714714	stavebnictvi@gmail.com	+420 784 145 145

ZPĚT

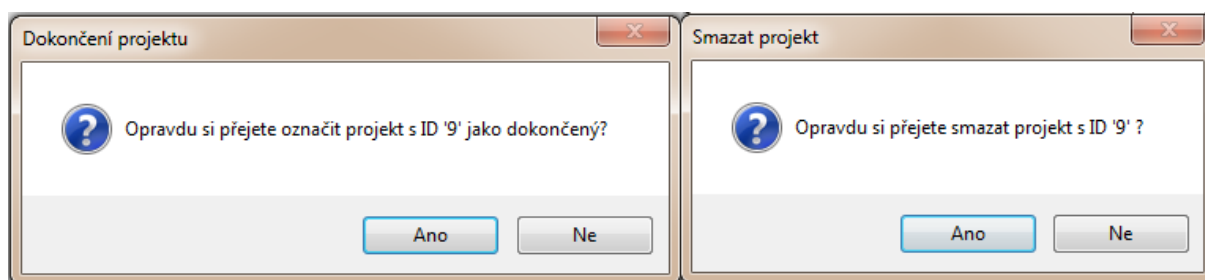
Obrázek 4.7: Formulářové zobrazení formuláře *frmProjekty*

V záhlaví formuláře je umístěno tlačítko *Nový projekt*, to po kliknutí otevře nový formulář, který slouží k přidání nového záznamu do tabulky *tblProjekty*. Textová pole v tomto formuláři odpovídají polím v tabulce *tblProjekty*. Nejdříve je nutno projektu navolit službu, zákazníka a zpracovatele. Tato volba probíhá pomocí předem nadefinovaných seznamů. Znamená to, že například při výběru zákazníka se zobrazí jen seznam zákazníků, kteří již jsou přidáni v tabulce *tblZakaznici*. Toto pomáhá předcházet problémům, aby nemohl být vytvořen projekt pro zákazníka, který neexistuje apod. Tento způsob zadávání analogicky platí i pro vyplňování služby a zpracovatele. Po vyplnění zbývajících textových polí a stisknutí tlačítka *Vytvořit* je zkontrolováno, zda je formulář korektně vyplněn. V případě bezchybného stavu je záznam přidán do tabulky *tblProjekty*. V případě chybného vyplnění vyskočí hláška upozorňující na daný nedostatek a kurzor je aktivován do textového pole, kde se daný nedostatek nachází.

Hlavní část formuláře je určena k zobrazování a procházení jednotlivých záznamů tabulky. Tyto záznamy jsou zobrazeny pomocí uzamčených textových polí. Toto uzamčení je opět z důvodu předcházení chyb, aby nebylo možno v tomto formuláři měnit záznamy projektů. Pod těmito poli se opět nachází navigační tlačítka pro přecházení mezi záznamy. Ve spodní části formuláře se nachází tři podformuláře. Ty slouží k zobrazení informací týkajících se daného projektu. Uživatel může kliknutím na záložku s názvem zobrazit informace o zákazníkovi, službě nebo zpracovateli projektu. Ve výchozím zobrazení jsou zobrazeny informace o zákazníkovi.

V pravé části formuláře se opět nachází vyhledávací textové pole, které vyhledá záznam projektu v tabulce *tblProjekty* podle jeho ID.

Dále se zde nachází tlačítka pro upravení, dokončení a smazání projektu. Tlačítko *Upravit projekt* otevře formulář sloužící k úpravě právě zobrazeného záznamu projektu. Tento formulář je opět velmi podobný formuláři pro nový záznam. Rozdíl je v tom, že textová pole formuláře pro úpravu obsahují údaje z formuláře projektů. V tomto formuláři je znemožněno upravovat ID projektu, protože se jedná o primární klíč a dále není možno upravovat datum zadání. Zadávání polí služby, zákazníka a zpracovatele je opět zajištěno poli s předdefinovaným seznamem, který je popsán výše. Při stisknutí tlačítka *Změnit* je opět zkontrolována validita vyplněných polí a v případě chyby je vyhozená hláška. Tlačítko *Dokončit projekt* označí daný projekt za dokončený a již nebude moci být dále upravován. Tlačítko *Smazat projekt* daný záznam projektu smaže z tabulky *tblProjekty*. Opět platí, že po kliknutí na tlačítko dokončení nebo smazání se zobrazí dialogové okno a žádá potvrzení pro dokončení akce (Obrázek 4.8).



Obrázek 4.8: Dialogová okna

Poslední částí popisu formuláře jsou tlačítka sloužící k tisku sestav. Tlačítko *Tisknout projekt* zobrazí sestavu, na které jsou veškeré informace o daném projektu včetně zadavatele a názvu služby. Dále tlačítko *Tisknout seznam projektů* zobrazí sestavu, která obsahuje seznam všech projektů. Tlačítko *Tisknout nedokončené projekty* zobrazí sestavu, ve které jsou jen ty projekty, které nejsou označeny jako dokončené.

Formulář frmSluzby

Zdrojem dat pro formulář *frmSluzby* (Obrázek 4.9) je především tabulka *tblSluzby*. Tento formulář slouží k zobrazování záznamu z této tabulky a k úpravám, přidávání či mazání záznamů z ní. Dále jsou na formuláři dostupná tlačítka pro tisk sestav, které souvisí se službami, které projekční kancelář nabízí. Formátování tohoto formuláře opět je velmi podobné ostatním formulářům, které pracují s daty tabulek.

ID	Jméno	Příjmení	Město	Ulice	Č. p.	PSČ	Zadáno	Cena	Odměna	Dokončeno
10	Jan	Malý	Ostrava	Nová	356	734 12	3.3.2015	12 000,00 Kč	550,00 Kč	
9	Jan	Malý	Paskov	Paskovská	13	174 51	26.2.2015		700,00 Kč	

Obrázek 4.9: Formulářové zobrazení formuláře *frmSluzby*

V záhlaví formuláře je opět tlačítko pro přidání nového záznamu do tabulky *tblSluzby*. Po stisknutí tlačítka *Nová služba* se otevře formulář. Tento formulář obsahuje především jen jedno textové pole, které musí být vyplněno a to název služby. Popis služby je volitelný a datum

přidání služby je generováno automaticky podle aktuálního data, tohle pole lze případně upravit. Po stisku tlačítka *Přidat* jsou zkontrolována textová pole, zda jsou správně vyplněná. Pokud jsou pole správně vyplněná, pak je přidán nový záznam do tabulky, v případě chyby je vyhozená hláška s chybovým hlášením.

Hlavní část formuláře, stejně jako u ostatních formulářů, slouží k zobrazování jednotlivých záznamů pomocí uzamčených textových polí. Ty jsou opět uzamčeny z důvodu předcházení chybovosti. Pod těmito textovými poli se již standardně nachází navigační tlačítka pro přechod mezi záznamy tabulky. Ve spodní části formuláře se opět nachází podformulář, tento zobrazuje všechny projekty, pro které se daná služba zpracovává.

Dále se na formuláři nachází tlačítka pro upravení a smazání služby. Tlačítko *Upravit tuto službu* otevře formulář sloužící k úpravě právě vybraného záznamu služby. V rámci formuláře pro úpravu je znemožněno upravit ID služby, jelikož jde o primární klíč záznamu v tabulce. Dále je znemožněno upravovat datum přidání služby. Tlačítko *Smazat tuto službu* slouží ke smazání právě vybraného záznamu z tabulky *tblSluzby*. Po stisku tlačítka je ještě zobrazeno dialogové okno žádající potvrzení smazání, aby nedošlo k neúmyslnému vymazání záznamu z databáze.

Nakonec se na formuláři nachází tři tlačítka pro tisk sestav. Tlačítko *Tisknout seznam služeb* zobrazí sestavu, na které je zobrazen seznam všech služeb. Tlačítko *Tisknout projekty* slouží pro zobrazení sestavy, kde je seznam všech projektů, pro které se právě zobrazená služba vykonává. A poslední tlačítko *Tisknout nedokončené projekty* zobrazí seznam nedokončených projektů, pro které se právě zobrazená služba zpracovává.

Formulář frmZpracovatele

Dále mezi hlavní formuláře patří formulář *frmZpracovatele* (Obrázek 4.10). Zdroj dat pro tento formulář je tabulka *tblZpracovatele*. V tomto formuláři se zobrazují data týkající se zpracovatelů projektu. Opět platí, že záznamy lze v přes tento formulář přidávat, upravovat či mazat a je dostupné tlačítko pro tisk sestavy.

Zpracovatelé

Nový zpracovatel

Rodné číslo:

971524/5410

Jméno:

Jan

Příjmení:

Novák

Mobil:

+420 748 414 145

E-mail:

novak@gmail.com

Bankovní účet:

000000-1425415411/2014

Město:

Ostrava

Ulice:

Nová

Číslo popisné:

4512

PSČ:

738 41

Upravit zpracovatele

Smazat zpracovatele

Tisknout seznam zpracovatelů

◀ ▶ ⏪ ⏩

Projekty

ID	Název	Město	Ulice	Číslo popisné	PSČ	Zadáno	Cena	Dokončeno
9	Plyn	Paskov	Paskovská	13	174 51	26.2.2015	17 500,00 Kč	
8	Elektřina	Ostrava	Nová	148	718 45	26.2.2015	1 700,00 Kč	26.2.2015
3	Projektování	Opava	Stromová	98	784 51	25.2.2015	24 500,00 Kč	25.2.2015

ZPĚT

Obrázek 4.10: Formulářové zobrazení formuláře *frmZpracovatele*

V záhlaví formuláře se nachází tlačítko *Nový zpracovatel*. Po kliknutí na toto tlačítko se otevře formulář, přes který se přidává nový záznam zpracovatele do tabulky *tblZpracovatele*. Na formuláři jsou textová pole, z těchto musí být především pole pro rodné číslo, jméno a příjmení vyplněno. Nový záznam je přidán do tabulky po kliknutí na tlačítko *Přidat*, opět platí, že v případě chybného vyplnění vyskočí chybová hláška, která oznámí, o jakou chybu se jedná.

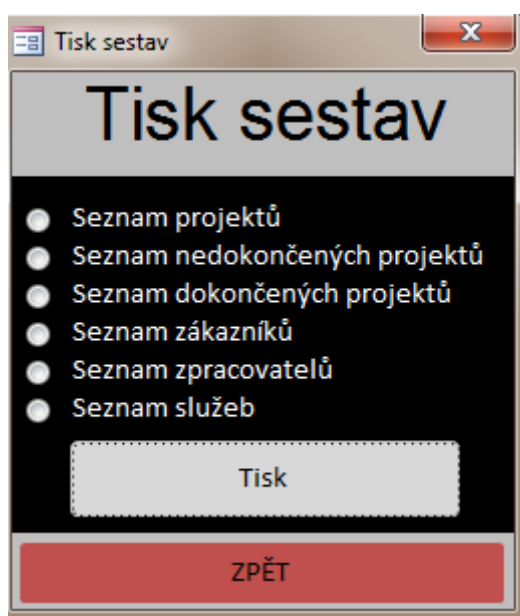
Hlavní část formuláře slouží, stejně jako u předcházejících formulářů, k zobrazování jednotlivých záznamů tabulky *tblZpracovatele*. Ty jsou zobrazeny v uzamčených textových polích. Opět platí, že uzamčení polí je z důvodu vyvarování se neúmyslným změnám dat v databázi. Pod textovými poli se již tradičně nachází navigační tlačítka pro přechod mezi jednotlivými záznamy v tabulce. Dolní část formuláře tvoří podformulář, který zobrazuje informace o projektech, které souvisí s právě vybraným zpracovatelem – tedy ty projekty, které daný zpracovatel zpracovává nebo zpracoval.

Další část formuláře tvoří tlačítka pro úpravu a smazání právě zobrazeného záznamu zpracovatele. Stisknutí tlačítka *Upravit zpracovatele* otevře formulář, ve kterém jsou textová pole, která obsahují data z hlavního formuláře. V rámci formuláře pro úpravu záznamu je znemožněno upravit rodné číslo zpracovatele, jelikož se jedná o primární klíč tabulky. Dále tlačítko *Smazat zpracovatele* vyvolá dialogové okno, které žádá o potvrzení, zda chce uživatel

opravdu vymazat záznam z tabulky. V případě potvrzení tohoto dialogového okna je provedeno vymazání záznamu. Posledním tlačítkem formuláře je tlačítko *Tisknout seznam zpracovatelů*. To po kliknutí otevře sestavu všech zpracovatelů, kteří ve firmě působí.

Formulář frmTisk

Posledním formulářem, který patří mezi ty hlavní, je formulář *frmTisk* (Obrázek 4.11). Tento formulář je slouží k tisku sestav na základě tabulek databáze. Vzhled formuláře je odlišný od ostatních, nezobrazuje přímo žádná data, pouze nabízí seznam sestav, které může zobrazit. I tento formulář je naformátován ve stejných barvách jako ty předchozí formuláře, aby se uživateli intuitivně a snadno s formulářem pracovalo.



Obrázek 4.11: Formulářové zobrazení formuláře *frmTisk*

Po kliknutí na tlačítko *Tisk* na úvodním formuláři (Obrázek 4.4) se otevře formulář pro tisk sestav. V hlavní části formuláře je zobrazen seznam sestav, které je možno vytisknout. Platí, že se jedná sestavy, které pracují se všemi záznamy v tabulkách. Tisk těchto sestav je přístupný i ve formulářích, které s nimi souvisí. Tedy například tisk seznamu projektů je dostupný i ve formuláři *frmProjekty* (toto analogicky platí i pro ostatní sestavy). Není tedy možné přes tento formulář vytisknout například sestavu pouze o jednom konkrétním projektu (to samé platí pro zákazníka, zpracovatele či službu).

4.3.4 Sestavy

Sestavy v aplikaci slouží k prezentaci dat z tabulek. V této aplikaci bylo vytvořeno mnoho sestav pro různé účely. Všechny sestavy byly vytvořeny v podobném formátování. Znamená

to, že rozložení dat na stránce je podobné. Dále byla z ekonomických důvodů použita na stránce pouze černá barva, aby nedocházelo při tisku většího objemu dokumentů ke zbytečně velkým nákladům na tisk.

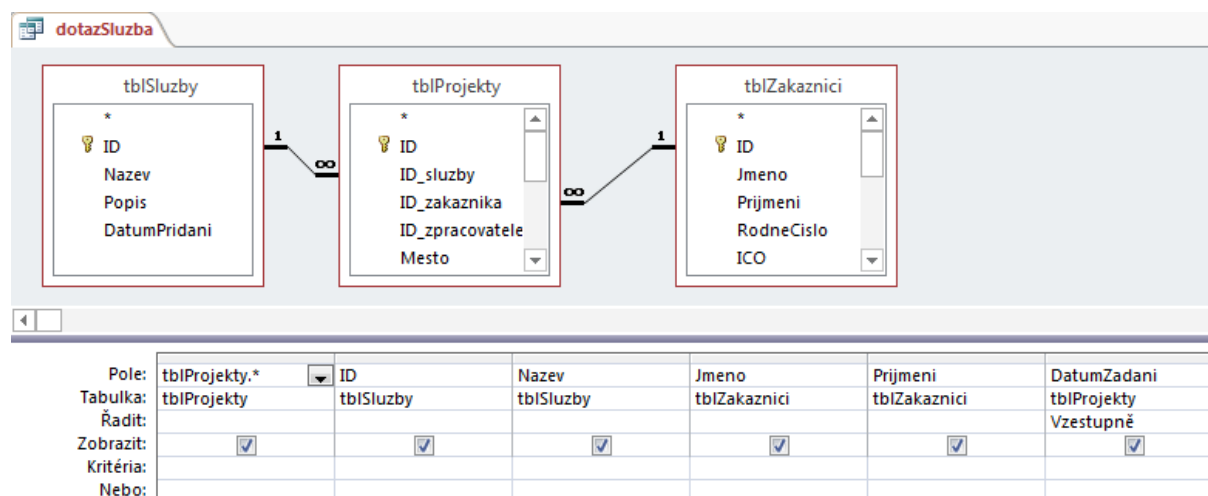
Obecně platí, že v rámci aplikace jsou dva druhy sestav. Prvním druhem jsou sestavy, které pracují s daty celé tabulky a prezentují tedy souhrnně data z celé tabulky. Tento druh sestav je možno zobrazit pomocí samostatného formuláře určeného pro tisk (Obrázek 4.11) nebo pomocí ostatních hlavních formulářů. Příkladem může být sestava všech projektů projekční kanceláře (Příloha 1).

Tím druhým jsou sestavy, které zobrazují data tabulek databáze související s některým záznamem. Zobrazení tohoto druhu sestavy je možné pouze z některého z hlavních formulářů, kde je zobrazen záznam, ke kterému se má daná sestava vázat. Příkladem pro tento případ může být zobrazení sestavy o konkrétním zákazníkovi (Příloha 2). V rámci této sestavy se zobrazují osobní informace zákazníka a navíc informace o jeho projektech.

4.3.5 Dotazy

V rámci aplikace bylo vytvořeno mnoho dotazů. Jsou využívány zejména jako výběrové dotazy, tedy jako zdroj záznamů pro formuláře či sestavy, aby v nich bylo možno zobrazit všechny potřebné údaje z tabulek.

Jako příklad těchto dotazů může být dotaz dotazSluzba (Obrázek 4.12). Ten je využíván jako zdroj záznamů v sestavě seznamu projektů pro určitou službu. Je důležité, aby se v této sestavě (Příloha 3) zobrazovaly kromě informací o projektu také informace o zákazníkovi a službě. Toho je pomocí tohoto dotazu úspěšně docíleno.



Obrázek 4.12: Návrhové zobrazení dotazu dotazSluzba

Druhou možností, kdy jsou dotazy využívány, je situace, kdy se uživatel snaží pomocí formuláře upravit, přidat nebo smazat záznam z tabulky. Tyto dotazy jsou součástí kódu jazyka VBA (Obrázek 4.13).

```
cmd.CommandText = "INSERT INTO tblSluzby(Nazev, Popis, DatumPridani) " & _  
"VALUES (?, ?, ?) "
```

Obrázek 4.13: Část dotazu pro vložení záznamu do tabulky tblSluzby

4.3.6 Dokončení aplikace

I přes to, že aplikace bude používána pouze na jednom počítači jedním uživatelem, je nutné provést některé kroky zabezpečující aplikaci. Toto je činěno za účelem, aby nemohlo dojít k nevědomému narušení základního fungování celé aplikace.

V aplikaci tedy došlo ke skrytí tlačítek a voleb, které by uživatel neměl při běžné práci s databázovou aplikací potřebovat. Konkrétně to je například znemožnění návrhového zobrazení či skrytí pásu karet tak, aby byl dostupná pouze karta *Domů*. V případě potřeby úpravy aplikace lze návrhové zobrazení a pásy karet opětovně zobrazit tak, že aplikace se spustí se stisknutou klávesou Shift.

4.4 Nasazení aplikace

Nasazení databázové aplikace proběhne tak, že samotná aplikace bude zaznamenána na paměťové médium. To bude předáno projekční kanceláři. Dále by neměl být s nasazením problém. Počítač, na kterém bude databázová aplikace provozována, plně vyhovuje hardwarovým nárokům aplikace. Na tomto počítači je také nainstalován balík programů Microsoft Office, který je nutný pro spuštění aplikace.

5 Závěr

Cílem bakalářské práce bylo navržení a implementace databázové aplikace pro projekční kancelář DK projekt s.r.o. sídlící v Ostravě. Aplikace bude firmě sloužit jako jednotný systém evidence dat, ve kterém bude schraňovat informace týkající se projektů, zákazníků, služeb a zpracovatelů.

Postup návrhu databázové aplikace probíhal v několika krocích. První krok byl zaměřen na teoretický základ potřebný při vytváření databázových aplikací. Byly zde rozebrány pojmy související s databázemi, dále zde byl podrobně popsán návrh databáze pomocí konceptuálního, logického a fyzického návrhu modelu. V další části byl popsán jazyk SQL a následoval popis pojmů a prostředí Microsoft Access.

Druhý krok byl zaměřen na analýzu současného stavu ve firmě, pro kterou byla výsledná databázová aplikace navrhnutá a vytvořena. V tomto kroku také byly určeny požadavky na výslednou aplikaci. Tato aplikace by je měla splňovat, aby vyhovovala projekční kanceláři a mohla tak nahradit současný systém, který je ve firmě používán.

Třetí krok souvisel s návrhem databázové aplikace. S využitím teoretických poznatků a informací získaných v rámci analýzy současného stavu ve firmě byl nejdříve vytvořen návrh konceptuálního modelu pomocí ER diagramu. Tento návrh byl dále převeden do návrhu logického modelu, kde byl vytvořen návrh tabulek a vztahů, které by měly být obsaženy v aplikaci. V poslední části byla vytvořena na základě výstupů z předešlých návrhů databázová aplikace. Jsou zde popsány tabulky a jejich struktura, vztahy mezi tabulkami. Dále následuje popis formulářů a sestav databázové aplikace.

Cíl bakalářské práce byl úspěšně splněn, kompletní databázová aplikace v prostředí Microsoft Access 2010 je připravena k užívání.

Je samozřejmostí, že během užívání aplikace firmou budou zjištěny nedostatky, které nebyly na první pohled při návrhu zřejmé. V budoucnu by tedy bylo možné aplikaci rozšířit či upravit tyto nedostatky tak, aby plně vyhovovala potřebám projekční kanceláře.

Seznam použité literatury

Literatura

HERNANDEZ, Michael J. *Návrh databází*. Praha: Grada, 2006. ISBN 80-247-0900-7.

KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech*. Praha: Ekopress, 2012. ISBN 978-80-86929-81-1.

KROENKE, David a David J AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

KRUCZEK, Aleš. *1001 tipů a triků pro Microsoft Access 2007-2010*. Brno: Computer Press, 2011. ISBN 978-80-251-3507-5.

KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. ISBN 978-80-251-3289-0.

OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

PÍSEK, Slavoj. *Access 2010: podrobný průvodce*. Praha: Grada Publishing, 2011. ISBN 978-80-247-3653-2.

SHEPHERD, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. ISBN 978-80-251-3686-7.

ŠIMONOVÁ, Stanislava a Jan PANUŠ. *Databázové systémy I*. Pardubice: Univerzita Pardubice, 2007. ISBN 978-80-7194-988-6.

Internetové zdroje

JAK NA OFFICE. *Datové typy a vlastnosti polí tabulek* [online]. JaknaOffice.cz, 20. 11. 2011 [cit. 2015-03-13]. Dostupné z: <http://www.jaknaoffice.cz/9-access/36-access-2010/41-datove-typy-a-vlastnosti-poli-tabulek/>

NOVÁK, Vítězslav. *Přednášky předmětu: Základy databází* [online]. 2014. [cit. 2015-03-05]. Dostupné z: <http://lms.vsb.cz/course/view.php?id=8933>

ŽÁRSKÁ, Zuzana. *MS ACCESS 2010 relační databáze* [online]. 2012. [cit. 2015-03-15]. Dostupné z: http://moodle2.voskop.eu/download/teu/U39_MS_Access.pdf

Seznam zkratek

1NF – první normální forma

2NF – druhá normální forma

3NF – třetí normální forma

4NF – čtvrtá normální forma

5NF – pátá normální forma

BCNF – Boyce Coddova normální forma

BOZP – Bezpečnost a ochrana zdraví při práci

CK – cizí klíč

DBMS – Database Management System

DCL – Data Control Language

DDL – Data Definition Language

DIČ – daňové identifikační číslo

DML – Data Manipulation Language

DQL – Data Query Language

ER – Entity-Relationship

IBM – International Business Machines Corporation

IČO – identifikační číslo osoby

IMS – Information Management System

PK – primární klíč

PSČ – poštovní směrovací číslo

SQL – Structured Query Language

SŘBD – Systém řízení báze dat

VBA – Visual Basic for Application

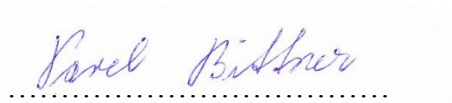
VBE – Visual Basic Editor

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 7. 5. 2015



Pavel Bittner

Seznam příloh

Příloha 1: Sestava všech projektů

Příloha 2: Sestava výpisu zákazníka

Příloha 3: Sestava seznamu projektů pro službu

Příloha 4: CD: soubor s databázovou aplikací

Seznam projektů

ID	Název	Jméno	Příjmení	Město	Ulice	Č. p.	PSČ	Zadáno	Cena	Odměna	Dokončeno
3	Projektování	Stavebnictví s.r.o		Opava	Stromová	98	784 51	25.2.2015	24 500,00 Kč		25.2.2015
5	Elektřina	Stavebnictví s.r.o		Písek	Písečná	458	714 11	26.2.2015	2 500,00 Kč		
6	Elektřina	Petr	Kachlíčka	Paskov	Nová	12	718 45	26.2.2015	15 800,00 Kč		
8	Elektřina	Petr	Kachlíčka	Ostrava	Nová	148	718 45	26.2.2015	1 700,00 Kč		26.2.2015
9	Plyn	Jan	Malý	Paskov	Paskovská	13	174 51	26.2.2015	17 500,00 Kč	700,00 Kč	
10	Plyn	Jan	Malý	Ostrava	Nová	356	734 12	3.3.2015	12 000,00 Kč	550,00 Kč	
11	Elektřina	Stavebnictví s.r.o		Ostrava	Rychlá	98	719 84	3.3.2015	14 000,00 Kč	300,00 Kč	

Výpis zákazníka

ID: 8
Jméno: Stavebnictví s.r.o
IČO: 14714714
DIČ: CZ1471471475
E-mail: stavebnictvi@gmail.com
Mobil: +420 784 145 145
Bakovní účet: 000000-1241212121/2010
Město: Opava
Ulice: Větmá
PSČ: 741 52

Projekty:

ID	Služba	Město	Ulice	Č. p.	PSČ	Zadáno	Cena	Odměna	Dokončeno
3	Projektování	Opava	Stromová	98	784 51	25.2.2015	24 500,00 Kč		25.2.2015
5	Elektřina	Písek	Písečná	458	714 11	26.2.2015	2 500,00 Kč		
11	Elektřina	Ostrava	Rychlá	98	719 84	3.3.2015	14 000,00 Kč	300,00 Kč	

Odměna celkem: 300,00 Kč

Seznam projektů

ID	Název	Jméno	Příjmení	Město	Ulice	Č. p.	PSČ	Zadáno	Cena	Dokončeno
8	Elektřina	Petr	Kachlíčka	Ostrava	Nová	148	718 45	26.2.2015	1 700,00 Kč	26.2.2015
6	Elektřina	Petr	Kachlíčka	Paskov	Nová	12	718 45	26.2.2015	15 800,00 Kč	
5	Elektřina	Stavebnictví s.r.o		Písek	Písečná	458	714 11	26.2.2015	2 500,00 Kč	
11	Elektřina	Stavebnictví s.r.o		Ostrava	Rychlá	98	719 84	3.3.2015	14 000,00 Kč	